

(19)



Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 0 796 475 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
28.07.1999 Bulletin 1999/30

(51) Int Cl.⁶: **G06T 7/00, G06K 9/38**

(86) International application number:
PCT/US95/15859

(21) Application number: **95942568.7**

(87) International publication number:
WO 96/18170 (13.06.1996 Gazette 1996/27)

(22) Date of filing: **28.11.1995**

(54) ADAPTIVE VISION SYSTEM USING DUAL THRESHOLDING

ADAPTIVES SICHTSYSTEM MIT DUALER SCHWELLWERTVERARBEITUNG

SYSTEME DE VISION ADAPTATIVE A DOUBLE SEUILLAGE

(84) Designated Contracting States:
**AT BE CH DE DK ES FR GB GR IE IT LI LU NL PT
SE**

(30) Priority: **05.12.1994 US 349212**

(43) Date of publication of application:
24.09.1997 Bulletin 1997/39

(73) Proprietor: **E.I. DU PONT DE NEMOURS AND
COMPANY**
Wilmington Delaware 19898 (US)

(72) Inventors:
• **VAIDYANATHAN, Akhileswar, Ganesh**
Hockessin, DE 19707-1142 (US)
• **WHITCOMB, James, Arthur**
Wilmington, DE 19810-3435 (US)

(74) Representative: **Jones, Alan John**
CARPMAELS & RANSFORD
43 Bloomsbury Square
London, WC1A 2RA (GB)

(56) References cited:

- **COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING, FEB. 1984, USA, vol. 25, no. 2, ISSN 0734-189X, pages 151-168, WERMSE R ET AL 'Segmentation of blood smears by hierarchical thresholding'**
- **SIGNAL PROCESSING, FEB. 1989, NETHERLANDS, vol. 16, no. 2, ISSN 0165-1684, pages 97-108, PAL N R ET AL 'Entropic thresholding' cited in the application**
- **1995 IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS. INTELLIGENT SYSTEMS FOR THE 21ST CENTURY (CAT. NO.95CH3576-7), 1995 IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS. INTELLIGENT SYSTEMS FOR THE 21ST CENTURY, VANCOUVER, ISBN 0-7803-2559-1, 1995, NEW YORK, NY, USA, IEEE, USA, pages 1888-1891 vol.2, VAIDYANATHAN A G ET AL 'Adaptive Image analysis for object recognition. I. Entropic object location'**

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 0 796 475 B1

Description**BACKGROUND OF THE INVENTION**

5 [0001] This application is a continuation-in-part of U.S. Application Serial No. 07/767,339, filed September 26, 1991, now U.S. Patent No. _____.

1. Field of the Invention

10 [0002] The present invention relates to image analysis methods and systems for automatically identifying objects in a background by multiply searching an image of the object and the background for a candidate object and validating the candidate object which has at least one predetermined attribute value of a valid object.

2. Description of the Related Art

15 [0003] Analyzing a representation of gray level space of an image has previously been proposed by D. Marr in an article entitled "Early Processing of Visual Information", Image Understanding, eds. S. Ullman and W. Richards, Ablex Publishing Company, 1984. The concept of entropy in information theory for signal processing was first proposed by Shannon in an article entitled "A Mathematical Theory of Communication", Bell System Technology J., Vol. 27, July 20 1948, pp. 379-423. Shannon showed that the entropy function:

$$25 \quad H(p_1, p_2, \dots, p_n) = - \sum_{k=1}^n p_k \ln p_k \quad (1)$$

uniquely satisfies the following three properties:

- 30 (1) $H(p_1, p_2, \dots, p_n)$ is a maximum for $p_k = 1/n$ for $k = 1, \dots, n$;
 (2) $H(AB) = H(A) + H_A(B)$, where A and B are two finite partitions and $H_A(B)$ is the conditional entropy of partition B given partition A;
 (3) $H(p_1, p_2, \dots, p_n, 0) = H(p_1, p_2, \dots, p_n)$

[0004] In addition, $H_{\max}(1/n, \dots, 1/n) = \ln n$.

35 [0005] The idea of using entropy to analyze a gray level histogram of an image was originally proposed by Pun in an article entitled "Entropic Thresholding, a New Approach", Comp. Graphics and Image Proc., Vol. 16, 1981, pp. 210-239. The entropy analysis of Pun was further refined by Kapur et al. in an article entitled "A New Method for Grey-Level Picture Thresholding Using the Entropy of the Histogram", Comp. Graphics and Image. Proc. 29, 1985, pp. 273-285. As shown by Pun and refined by Kapur, the concept of entropy can be extended to two dimensions if the 40 gray level histogram of the image is used to define a probability distribution:

$$p_s = f_s / N \text{ for } s=1, \dots, N_{\text{gray}} \quad (2)$$

45 where

f_s = frequency of gray level s
 N = # pixels in image
 N_{gray} = # gray levels

50

It follows that the entropy function of a histogram describing an image with a uniform gray level distribution is at a maximum. The more peaks in the distribution, the lower the entropy.

[0006] Pal and Pal, in an article entitled "Entropic Thresholding", Signal Processing, Vol. 16, 1989, pp. 97-108, recognized that a gray level histogram does not contain information regarding the spatial relationship between the gray levels and suggested the use of a two-dimensional data structure as an alternative to the gray level histogram. The structure suggested is a Haralick co-occurrence matrix as described by Haralick et al. in "Textural Features for Image Classification", IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-3, No. 6, 1973, pp. 610-621.

[0007] The techniques described above for analyzing the gray level space of an image compute a single entropic threshold and do not contemplate a system for recursively analyzing the gray level space of the image by automatically computing multiple entropic thresholds. Moreover, the above-discussed articles are not concerned with object validation and do not propose a driver for object comparison and validation.

5 [0008] In an article entitled "Automatic Multithreshold Selection", Computer Vision Graphics and Image Processing, Vol. 25, 1984, pp. 46-67, Wang and Haralick describe an attempt at multi-threshold selection using gray level distributions in local neighborhoods of each pixel in an image. This technique does not employ image entropy. Rather, it adaptively computes the local background of an object as an image is scanned by a priori selection of a gray level neighborhood. Moreover, Wang and Haralick apply multiple thresholds to process the image, which results in the labeling of regions of the image. Wang and Haralick do not use multiple thresholds in order to perform object identification and validation.

10 [0009] Automated bacterial colony counting systems are commercially available for determining the number of visible bacteria in or on a culture medium such as a Petri dish. Examples include a Domino Image Analysis System made by Perceptive Instruments (Essex, England), a Seescan Imaging Plate Reader made by Seescan Imaging Limited (Cambridge, England), a Protos Colony Counter and Image 40-10 Analyzer, made by Analytical Measuring Systems (Cambridge, England), Bio-Foss Colony Counting System made by Foss Electric (York, England), an Artek 810 Image Analyzer made by Dynatech Laboratories, Inc. (Chantilly, VA), an Optimas Image Analyzer made by Bio Scan (Edmonds, WA), a Video Densitometer II made by Biomed Instruments, Inc. (Fullerton, CA), and an Image-Pro Plus made by Media Cybernetics (Silver Spring, MD). All of these instruments require the input of a single predetermined threshold per each image. None are capable of using image entropy to recursively compute multiple thresholds for object identification.

15 [0010] A semi-automated system for detection and quantification of microbial growth in sections of food has been described by Fernandes et al. in "Detection and Quantification of Microorganisms in Heterogeneous Foodstuffs by Image Analysis", Cabios, Vol. 4, No. 2, 1988, pp. 291-295. The system described in this article relies on manually examining the gray level histogram to identify a single threshold.

20 [0011] A number of patents have issued on colony counting systems, such as U.S. Patent No. 4,637,053 to Schalkowsky, U.S. Patent No. 3,811,036 to Perry, U.S. Patent No. 3,736,432, and French Patent Application No. 2,602,074 to Segui et al. None of these patents discloses the concept of using image entropy to adaptively segment an image for colony enumeration and identification. "Adaptively segmenting" means thresholding the image with more than one threshold gray level.

25 [0012] Segmentation methods based on histogram analysis and threshold finding are disclosed in "Computer Vision, Graphics and Image Processing", Vol. 25, No. 2, Feb. 1984, NY (US).

[0013] Accordingly, it would be desirable to provide methods and systems which use multiple thresholds and perform object identification and validation. In one implementation, the threshold is an entropically selected threshold gray level to search an image for automatic object identification.

30 [0014] It would further be desirable to provide methods and systems which use entropically selected threshold gray levels to recursively search an image for automatic object identification.

[0015] It would further be desirable to provide a method and a system for automatic object identification which can be used in colony counting, colony screening, identification of discrete features in carpets and identification of pigment elements embedded in a polymer.

35 [0016] Additional advantages of the invention will be set forth in the description which follows, and in part will be obvious from the description, or may be learned by practice of the invention. The objects and advantages of the invention may be realized and obtained by means of the instrumentalities and combinations particularly pointed out in the appended claims.

40 SUMMARY OF THE INVENTION

[0017] In accordance with a first aspect of the present invention there is provided a method of identifying at least one valid object having at least one predetermined attribute value, as defined in Claim 1.

45 [0018] In accordance with a second aspect of the present invention there is provided another such method as defined in Claim 9.

[0019] The method of the present invention is best suited for automatic identification of fairly simple objects in a complex environment. In such situations, interior regions of the objects close to the boundaries (i.e., edge regions) are usually fairly uniform in gray level. The recursive nature of the method of the present invention makes it an adaptive technique, well-suited for searching widely varying backgrounds. For cases of identification of a complex object in a relatively simple environment, it may be necessary to recursively partition a co-occurrence matrix.

50 [0020] The method of the present invention can be applied to on-line industrial inspection techniques to enhance quality control in manufacturing processes where the product can be described fairly simply, but the environment of

the product is unknown.

BRIEF DESCRIPTION OF THE DRAWINGS

- 5 [0021] The accompanying drawings, which are incorporated in and constitute a part of the specification, illustrate the presently preferred embodiments of the invention and, together with the general description given above and the detailed description of the preferred embodiments given below, serve to explain the principles of the invention.
- [0022] Fig. 1 is a block diagram showing the steps of the overall method according to a first embodiment of the present invention.
- 10 [0023] Fig. 2 is a block diagram showing the steps of the overall method according to a specific implementation of the first embodiment of the present invention.
- [0024] Fig. 3 is a flow chart illustrating the steps of a module HISTOGRAM which is used to generate a gray level histogram of an image.
- [0025] Fig. 4 is a flow chart illustrating the steps of a module ENTROPY which is used to entropically select a threshold gray level such that the entropy function of the histogram is maximized.
- 15 [0026] Fig. 5 is a gray level histogram of an image of a single, simple object in a varying background.
- [0027] Fig. 6 is a flow chart illustrating the steps of a module SEARCH IMAGE which is used to search the image for at least one candidate object.
- [0028] Fig. 7 is a flow chart illustrating the steps of a module FIND OBJECT which is also used to search the image.
- 20 [0029] Figs. 8A and 8B comprise a flow chart illustrating the steps of a module TRACE OBJECT which is used to trace and validate the candidate object.
- [0030] Fig. 9 is a flow chart illustrating the steps of a module CHK GRAY which is used to detect whether the candidate object is relatively lighter or darker than the background.
- [0031] Figs. 10A - 10C are original, upper and lower gray level histograms, respectively, of an image.
- 25 [0032] Fig. 11 is a flow chart illustrating the steps of a module ANALYZE which is used to recursively search the image for candidate objects.
- [0033] Fig. 12 is a flow chart showing the steps of a module which uses entropic thresholding and generates a fixed threshold list from which to select desirable threshold pairs.
- [0034] Figs. 13A - 13D comprise a flow chart illustrating the steps of a module CHK LIST which is used to resolve redundancies in inhomogeneous objects.
- 30 [0035] Fig. 14 is a flow chart illustrating the steps of a module SET STAT which is used with the module CHK LIST as shown in Figs. 13A - 13D.
- [0036] Figs. 15A - 15B comprise a flow chart illustrating the steps of a module CHK LIST which is used to resolve redundancies in homogeneous objects.
- 35 [0037] Fig. 16 is a flow chart illustrating the steps of a module SET STAT which is used with the module CHK LIST as shown in Figs. 15A - 15B.
- [0038] Figs. 17A - 17B comprise a flow chart illustrating the steps of a module FINAL CHK which is used to perform a final check to resolve redundancies in inhomogeneous and homogeneous objects.
- [0039] Fig. 18 is a flow chart illustrating the steps of a module INT STAT which is used with the module FINAL CHK as shown in Figs. 17A - 17B.
- 40 [0040] Fig. 19A is a flow chart illustrating the overall steps of a module which is CALCON used to filter inhomogeneous objects.
- [0041] Fig. 19B is a flow chart illustrating the steps of the part of CALCON as shown in Fig. 19A which applies a heavy filter.
- 45 [0042] Fig. 19C is a flow chart illustrating the steps of the part of CALCON as shown in Fig. 19A which applies a medium filter.
- [0043] Fig. 19D is a flow chart illustrating the steps of the part of CALCON as shown in Fig. 19A which applies a light filter.
- [0044] Figs. 20A - 20B comprise a flow chart illustrating the steps of a module CALCON used to filter homogeneous objects.
- 50 [0045] Figs. 21A - 21D comprise a flow chart illustrating the steps of a module UNCLUMP which is used to unclump inhomogeneous objects.
- [0046] Figs. 22A - 22B comprise a flow chart illustrating the steps of a module PROCESS CLUMP which is used to process a clump of inhomogeneous objects.
- 55 [0047] Fig. 23A is an outline of a candidate clump.
- [0048] Fig. 23B is a distance buffer comprising a plurality of distance values representing the distance of each perimeter point of the candidate clump to the center of mass of the candidate clump shown in Fig. 23A.
- [0049] Fig. 23C is the distance buffer shown in Fig. 23B which has been shifted by a distance.

[0050] Fig. 24 is a block diagram showing the components of the system of the first embodiment of the present invention.

[0051] Fig. 25 is a block diagram showing the steps of the overall method of a second embodiment of the present invention.

5 [0052] Fig. 26 is a flow chart illustrating the steps of modules VERT, HORIZ, LFT DIAG and RT DIAG which are used to generate a co-occurrence matrix of an image according to the second embodiment.

[0053] Fig. 27 is a flow chart illustrating the steps of the module ENTROPY used with the second embodiment of the present invention.

[0054] Fig. 28 is a co-occurrence matrix which is partitioned into four quadrants.

10 [0055] Fig. 29 is a flow chart illustrating the steps of the module ANALYZE used with the second embodiment of the present invention.

[0056] Fig. 30 is a block diagram showing the steps of the overall method of a preferred embodiment of a third embodiment.

15 [0057] Fig. 31 is a gray level histogram which has been divided into $N + 1$ sub-histograms in accordance with the preferred embodiment of the third embodiment of the present invention.

[0058] Fig. 32 is a block diagram showing the steps of the overall method of a preferred implementation of a fourth embodiment of the present invention.

[0059] Fig. 33 is a block diagram of the system of the present invention in accordance with the preferred implementation of the fourth embodiment of the present invention.

20 [0060] Fig. 34 is a block diagram showing the steps of the overall method of a preferred implementation of a fifth embodiment of the present invention.

[0061] Fig. 33 is a block diagram showing the steps of the overall method of a preferred implementation of a sixth embodiment of the present invention.

25 [0062] Fig. 34 is a block diagram showing the steps of the overall method of a preferred implementation of a seventh embodiment of the present invention.

[0063] Fig. 35 is a block diagram showing the steps of the overall method of a preferred implementation of an eighth embodiment of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

30 [0064] Reference will now be made in detail to the present preferred embodiments of the invention as illustrated in the accompanying drawings.

[0065] In accordance with a first embodiment of the present invention, there is provided a method of identifying at least one valid object in a background. Fig. 1 is a block diagram showing the overall method of the present invention.

35 The method of the first embodiment is also referred to as recursive entropic thresholding in natural analysis. [0066] The method of the present invention comprises the step of generating an image of the object and the background. An image is generated as shown in block A of Fig. 1 and in block A of Fig. 2. Fig. 2 illustrates a preferred implementation of the method of the first embodiment, as illustrated in Fig. 1. The hardware used to implement the method of the present invention must first be initialized when the image is generated. The image of the object and the background may be generated by a camera. Although a CCD camera is generally used with the present invention, any type of camera may be used without departing from the general principles of the present invention. The image is then digitized and stored by a frame grabber or a video digitizer.

40 [0067] The method of the present invention also comprises the step of defining a data space representative of the image comprising a plurality of sub-spaces. This step is shown in block B of Fig. 1. The data space may be based on at least one predetermined attribute value of a valid object. Alternatively, the data space may be based on at least one predetermined attribute value of a previously identified object. Moreover, the data space may comprise the gray level values of the image. A histogram is an example of a one-dimensional data space constituting the gray level values associated with each point in the image. This example will be explained in more depth as a preferred implementation below. Alternatively, the data space may comprise a color space of the image. This color space may be the R,G,B color space, or the LAB color space. Such color spaces are examples of three-dimensional data spaces. Alternatively, the data space may comprise a space resulting from the transformation of gray level values of an image. This resulting space may be, for example, contrast, hue magnitude or edge intensity. The data spaces described in this paragraph are meant to be exemplary only, with the scope of the present invention extending beyond these examples.

45 [0068] In the preferred implementation as illustrated in Fig. 2, this step comprises generating a gray level histogram of the image, where the gray level histogram has an entropy function. This step is shown in block B of Fig. 2. A module, HISTOGRAM, is used to generate the gray level histogram of the region of interest of the image. The steps for generating the gray level histogram are shown in the flow of Fig. 3. As shown in block A of Fig. 3, HISTOGRAM first calculates a histogram of the region of interest of the image. It then calculates the values to be used subsequently in the calculation

of the entropy function, H_s , for each gray level, s , as shown in block B of Fig. 3. The results of this calculation are stored in a global look-up table as shown in block C. This ensures that for subsequent calculations of the entropic threshold gray level, only a simple look-up operation is required.

[0069] The method of the present invention also includes the step of selecting at least one sub-space. This step is shown in block C of Fig. 1. The sub-space is a bounded portion of the data space - i.e., it does not span the entire data space. The sub-space may be a pair of gray level values, or a range of gray level values. Moreover, the sub-space may be selected based on the way that for instance, gray level values, or color parameters, cluster in the respective spaces, although it should be noted that the sub-space is not limited to these examples. In the preferred implementation as illustrated in Fig. 2, this step comprises entropically selecting a threshold gray level such that the entropy function of the histogram is maximized, as shown in block C of Fig. 2. This step is performed by the ENTROPY module as shown in Fig. 4. As shown in block A of Fig. 4, the first step in maximizing the entropy function of the histogram is to initialize the maximum entropy function to a minimum value.

[0070] In a preferred implementation of the first embodiment, the step of entropically selecting a threshold gray level includes the sub-step of sequentially partitioning the gray level histogram at each gray level into a first partition and a second partition. To illustrate the simple case where a single, simple object in a varying background is identified, a gray level histogram of an image is shown in Fig. 5. The first and second partitions are shown in the histogram of Fig. 5, where the gray levels of the background are represented by a first partition A, and the gray levels of the valid object are represented by a second partition B. In the ENTROPY module the partitioned threshold gray level is initialized to a minimum value as shown in block B of Fig. 4.

[0071] The step of entropically selecting a threshold gray level also includes the sub-step of computing the entropy function for each partition, where the total entropy function of the histogram is defined as the sum of the entropy function $H_s(A)$ of first partition, A, and the entropy function $H_s(B)$ of second partition. This step is shown in block C of Fig. 4 and can be mathematically expressed as follows:

[0072] For a given threshold gray level, s ,

$$H_s(A) = - \sum_{i=1}^s \frac{P_i}{P_s} \ln \left(\frac{P_i}{P_s} \right) \quad (3)$$

with

$$P_i = \frac{f_i}{N_s} \quad P_s = \frac{1}{N} \sum_{i=1}^s f_i \quad (4)$$

Thus,

$$\frac{P_i}{P_s} = \frac{f_i}{\sum_{i=1}^s f_i} = \frac{f_i}{N_s} \quad \text{with } N_s = \sum_{i=1}^s f_i \quad (5)$$

So,

$$H_s(A) = - \sum_{i=1}^s \frac{f_i}{N_s} \ln \left(\frac{f_i}{N_s} \right) = - \frac{1}{N_s} \sum_{i=1}^s f_i \ln f_i + \ln N_s \quad (6)$$

Similarly,

$$H_s(B) = \frac{-1}{N_s'} \sum_{i=s+1}^{N_{\text{gray}}} f_i \ln f_i + \ln N_s' \quad (7)$$

where

$$N_s' = N - N_s \text{ and}$$

$$H_s(A) + H_s(B) = \ln N_s = \ln N_s' - \frac{1}{N_s} \sum_{i=1}^s f_i \ln f_i - \frac{1}{N_s'} \sum_{i=s+1}^{N_{\text{gray}}} f_i \ln f_i \quad (8)$$

[0073] The sum $H_s(A) + H_s(B)$ represents the total entropy function of the gray level histogram of the image. The maximum entropic threshold gray level is the value of s which maximizes the total entropy function.

[0074] Decision diamond D of Fig. 4 asks whether the entropy function of the histogram is greater than the maximum entropy function as initialized in block A. If it is, then the maximum entropy function is updated using the partitioned threshold gray level as shown in block E of Fig. 4. The maximum entropic threshold gray level is then set to the partitioned threshold gray level as shown in block F. After the maximum entropy threshold gray level has been set, or if the entropic function of the histogram is not greater than the maximum entropy function, then decision diamond G of the ENTROPY module as illustrated in Fig. 4 asks whether the partitioned threshold gray level equals the maximum threshold gray level. If so, the maximum entropic threshold gray level is returned as shown in block H of Fig. 4. If not, then the partitioned threshold gray level is incremented as illustrated in block I of Fig. 4, and the incremented partitioned threshold gray level is returned to block C, where the entropy function of the incremented, partitioned threshold gray level is computed. The loop through C - G is repeated until the partitioned threshold gray level equals the maximum threshold gray level, at which point the maximum entropic threshold gray level is returned as shown in block H.

[0075] According to the present invention, in $H_s(A)$ and $H_s(B)$, the probability distributions are renormalized to include only the gray levels within each of the partitions. With this renormalization, the maximum entropy function occurs right at the edge of the object peak in the gray level histogram as shown at T in Fig. 5. Thus, a new threshold gray level is selected such that the entropy function of the histogram is maximized. With this maximum choice of threshold for the simple case as illustrated in Fig. 5, the renormalized distribution of the background becomes the least peaky and the most uniform. The total entropy function of the histogram is dominated by the entropy function of the background, since the number of gray levels in the background partition is much larger than the number of gray levels in the object partition.

[0076] The method of the present invention also includes the step of multiply searching the image for at least one representation of a candidate object using each selected sub-space. This step may comprise either scanning the image once using each of the sub-spaces simultaneously, or scanning the image multiple times using a selected sub-space for each scan. In a preferred implementation of the first embodiment as illustrated in Fig. 2, this step comprises searching the image for at least one candidate object, wherein the candidate object has at least one candidate object attribute value. The searching step includes the sub-steps of scanning the image for at least one candidate object using the entropically selected threshold gray level and tracing the candidate object having boundary gray levels determined by the entropically selected threshold gray level. The searching step is performed by a module SEARCH IMAGE as shown in Fig. 7, a module FIND OBJECT of Fig. 8, and a module TRACE OBJECT as shown in Fig. 9.

[0077] The method of the present invention also includes the step of validating the candidate object having the valid object predetermined attribute value to identify the valid object. The validating step includes the sub-steps of calculating the candidate object attribute values and comparing the candidate object attribute values to the valid object predetermined attribute values to validate candidate objects. The calculating sub-step further includes the sub-step of storing the candidate object attribute values. The validating step is performed by the TRACE OBJECT module. In the first embodiment of the present invention, TRACE OBJECT uses only size and shape factor as valid object predetermined attribute values. In general, other attribute values may be used for the valid object predetermined attribute values.

[0078] The present invention employs a driver and a kernel. The driver stores the attribute values of the valid object, where each value represents the definition of a valid object, e.g., edge contrast, area, shape, etc. The driver of the present invention is specific to a given application. In an object-oriented environment, it is straight-forward in many instances to describe an object via a list of attributes such as size, shape, color, etc. For more complex objects where a simple parametric description might not be possible, one could use a neural network in the driver to identify the object.

Parameters derived from the candidate object can be fed into the neural network, which has been trained to recognize specific objects. At this point, the architecture of the present invention begins to resemble a neural vision architecture where there is a feedback loop between the brain and the eye. In the present invention, a high-order driver is intertwined with a lower-order kernel. In this case, a more complex description of the object in the driver is used to drive the searching process, which in turn identifies further candidate objects.

[0079] The driver drives the kernel. The kernel performs several functions. It calculates an entropically selected threshold gray level, searches the image and calculates the attribute values for a candidate objects. In addition, it performs a validity check on the candidate objects by comparing the attribute values of the candidate objects with the predetermined attribute values for the valid objects, which, as noted above, are contained in the driver. It also performs a redundancy check to prevent multiple identification of a valid object.

[0080] As illustrated by block A in Fig. 6, the first step in the SEARCH IMAGE module is to blacken a portion of the image not being searched. As shown in block B, the search position is then initialized. The module SEARCH IMAGE searches the region of interest with the current entropically selected threshold gray level. Decision diamond C of Fig. 6 then asks whether the search position is at the end of the scan. If so, then a module CHK GRAY, which is shown in Fig. 8 in detail and which will be described in greater detail below, is run as illustrated in block D. CHK GRAY retains only objects which are lighter than the background. To identify objects darker than the background, the image is inverted immediately after it has been generated. This allows CHK GRAY to retain objects which are darker than the background. Also, a module CHK LIST, which is shown in Figs. 13A - C and Figs. 15A and 15B in detail and which also prevents multiple identification of a valid object, is run as illustrated in block E. The number of new valid objects found by SEARCH IMAGE is returned as illustrated in block F of Fig. 6.

[0081] If the search position is not at the end of the scan, then the module SEARCH IMAGE searches the region of interest with the current entropically selected threshold gray level until it finds a point which has a gray level exceeding the entropically selected threshold gray level using a module FIND OBJECT. Such a point might be the first point of a new candidate object. Decision diamond G of Fig. 6 asks whether a new candidate object has been found using the module FIND OBJECT. If so, FIND OBJECT checks to see if the object has already been traced in the current search. If the object has not already been traced in the current search, the module SEARCH IMAGE proceeds to trace the object by running the module TRACE OBJECT, which is shown in detail in Fig. 8 as shown by block H of Fig. 6. After the object has been traced, the search position is incremented as illustrated in block I of Fig. 6. The loop through B - I is continued until the module SEARCH IMAGE is at the end of the search as indicated by decision diamond C.

Alternatively, if a new candidate object has not been found as indicated by decision diamond G, then the search position is incremented as illustrated in block I, thus by-passing the tracing step and the loop through C - I is continued until SEARCH IMAGE is at the end of the search.

[0082] The steps of the module FIND OBJECT are illustrated in Fig. 7. The first step in FIND OBJECT is to initialize the search position to the current location of the image being searched as shown in block A. Decision diamond B then asks whether the search position is inside the object. If so, then the search position is incremented as illustrated by block C, and decision diamond D asks whether FIND OBJECT is at the end of its search. If so, then no new object is found as indicated in block E. If not, then decision diamond B asks whether the incremented search position is inside the object. This process of looping through B - E continues until the search position is not inside the object. At this point, decision diamond F asks whether a next object has been found. If not, then the search position is incremented as illustrated in block G of Fig. 7, and decision diamond H asks whether the SEARCH IMAGE is at the end of the search. If so, then no new object found is returned as indicated by block I. If not, then decision diamond F again asks whether a next object has been found using the incremented search position. This process of looping through F - I continues until a next object has been found. Decision diamond J asks whether the object which has been found has already been traced. If so, then no new object found is returned as indicated by block K. If the object which has been found has not already been traced, then the search position is updated as illustrated by block L, and a new object found is returned as indicated by block M of Fig. 7.

[0083] The steps of the module TRACE OBJECT are illustrated in Figs. 8A and 8B. The basic principles of the TRACE OBJECT module are similar to those described in "Digital Image Processing" by Rafael C. Gonzalez and Paul Wintz, Second Ed., Addison-Wesley Publishing Company, Reading, Massachusetts (1987). As shown in block A of Fig. 8A, the first step in the TRACE OBJECT module is to initialize the candidate object attribute values. The TRACE OBJECT module then asks in decision diamond B whether a neighboring perimeter point has been found. If not, the traced object is invalid as illustrated by block C. If the neighboring perimeter point has been found, then decision diamond D asks whether the TRACE OBJECT module is at the first perimeter point of the candidate object. If not, then the candidate object attribute values are updated as illustrated in block E of Fig. 8A. The loop through B - E is then repeated using the updated candidate object attribute values until the TRACE OBJECT module is at the first perimeter point of the candidate object. The center of mass coordinate is then calculated as shown in block F of Fig. 8A. Decision diamond G then asks if the candidate object area is too large. If it is, the traced object is invalid as indicated by block H of Fig. 8A.

[0084] If the candidate object area is not too large, then a shape factor is calculated as shown in block I in Fig. 8B.

The definition of the shape factor may vary, depending on the geometry of the object being identified. For instance, the definition of the shape factor for circular objects is:

$$\text{Shape Factor} = 1 - \frac{p^2}{4\pi A} \quad (9)$$

Where:

- 10 p is the perimeter of a candidate object; and
 A is the area of the candidate object.

TRACE OBJECT then checks if the shape factor is within a predetermined range as contained in the driver as shown in decision diamond J in Fig. 8B. If the shape factor does not fall within the predetermined range, then the traced object is invalid as illustrated by block K of Fig. 8B. If the shape factor falls within the predetermined range, then the candidate object is added to the list of valid objects maintained by the kernel as shown by block L.

[0085] After all the candidate objects have been traced in the current search, the module CHK GRAY as shown in Fig. 9 is called to check whether the candidate objects are relatively lighter than the background. As shown in block A of Fig. 8, the first step in the CHK GRAY module is to advance to the first candidate object found in the current search. Decision diamond B of Fig. 9 asks whether the candidate object is the last object in the list of candidate objects. If it is, the module stops running as shown by oval C. If the candidate object is not the last object in the list of candidate objects, then the average exterior gray level is calculated as illustrated by block D. Decision diamond E then asks whether the gray level of the center of mass is greater than average exterior gray level of the four exterior points (i.e., the top, bottom, left and right points) surrounding the extremum points of the object. The exterior points are those points in the background which are immediate neighbors to the extremum points of the object. If not, the object is deleted as shown in block F of Fig. 9. If the gray level center of mass is greater than the average exterior gray level, then the candidate object is retained and the CHK GRAY module advances to the next candidate object as shown in block G. The CHK GRAY module then returns to decision diamond B to ask whether the candidate object is the last object. The loop as shown in B - G is repeated for the next candidate object until the next candidate object is the last candidate object, at which point CHK GRAY stops running. As noted above, the module CHK GRAY may be run to detect objects darker than the background. In this case, the image is initially inverted prior to performing the step of generating the image of the object and the background.

[0086] The method as described thus far can be referred to as a screening process. For example, it can be used to screen for the presence of pathological bacteria in food or in blood or soil samples. A screening process results in a yes - no answer; absolute quantitation is not necessary. For a more stringent identification process, it is necessary to implement the method of the present invention recursively as described below.

[0087] The method of the present invention further includes the steps of subdividing the data space into an upper sub-space as an upper delimiter and a lower sub-space as a lower delimiter and multiply searching by recursively repeating the selecting, validating and subdividing steps for each of the upper and lower sub-spaces, thereby recursively partitioning the data space until the condition for terminating the multiple searching has been reached. The terminating condition may be that a predetermined minimum number of new valid objects is identified. This predetermined minimum number may be greater than zero, or may be zero. Alternatively, the terminating condition could be a minimum gray level partition width.

[0088] In the preferred implementation of Fig. 2, the method of the present invention further comprises the steps of subdividing the gray level histogram into an upper histogram and a lower histogram using the entropic threshold gray level which was selected to maximize the entropy function of the histogram as an upper delimiter and a lower delimiter. The selection, searching, validating and subdividing steps are recursively repeated for each of the upper and lower histograms. The repetition of the selection step selects a next successive entropic threshold gray level, thereby recursively partitioning the gray level histogram to identify the valid objects until a predetermined minimum number of new valid objects is identified. In the preferred implementation, the predetermined minimum number is zero. However, there may be cases where the predetermined number is greater than zero, such as when a complete identification is not required.

[0089] Figs. 10A - 10C illustrates the concept of subdividing a histogram into an upper histogram and a lower histogram. An original histogram is shown in Fig. 10A. THRESH, as shown at T in Fig. 10A, is the entropically selected threshold gray level for the gray level histogram corresponding to the gray level region between the minimum gray level being searched and the maximum gray level being searched. For the original histogram as shown in Fig. 10A, the minimum gray level being searched is zero and the maximum gray level being searched is MAX. THRESH II, as shown at B, is the entropically selected threshold gray level for the gray level histogram corresponding to the gray level

region between THRESH and MAX. THRESH LO, as shown at A, is the entropically selected threshold gray level for the gray level histogram corresponding to the gray level region between zero and THRESH.

[0090] According to the present invention, the subdividing, selection, searching and validating steps are then recursively repeated. By recursion is meant the process of continuously dividing a histogram into upper and lower histograms, searching each upper histogram, which upper histogram is itself continuously divided into upper and lower histograms, for new valid objects until the number of new valid objects found in an upper histogram is less than or equal to a predetermined minimum number, and subsequently searching each lower histogram corresponding to the most recently searched upper histogram, which lower histogram is itself continuously divided into upper and lower histograms, until the number of new valid objects found in a lower histogram is less than or equal to the predetermined minimum number.

[0091] The upper histogram is shown in Fig. 10B. The repetition of the subdividing step subdivides the upper histogram into a next successive upper and lower histogram as shown in Fig. 10B. The repetition of the selection step for the upper histogram selects a next upper successive entropic threshold gray level, as shown at B in Fig. 10B. Thus, point B, which was THRESH HI in the original histogram, becomes the threshold for the upper histogram, or NEXT UPPER THRESH. In Fig. 10B, the minimum gray level being searched is now THRESH and the maximum gray level being searched is now MAX. The NEXT UPPER THRESH HI, shown at C, is the entropically selected threshold gray level for the gray level histogram corresponding to the gray level region between B and MAX. The NEXT UPPER THRESH LO, shown at D, is the entropically selected threshold gray level for the gray level histogram corresponding to the gray level region between THRESH and B. The selection, searching and validating steps are then repeated recursively using the next upper successive entropic threshold gray level, B, as the entropic threshold gray level.

[0092] Fig. 10C shows the lower histogram. The repetition of the subdividing step subdivides the lower histogram into a next successive upper and lower histogram as shown in Fig. 10C. The repetition of the selection step for the lower histogram selects a next lower successive entropic threshold gray level, as shown at A in Fig. 10C. Thus, point A, which was THRESH LO in the original histogram, becomes the threshold for the partitioned lower histogram, or NEXT LOWER THRESH. In Fig. 10C, the minimum gray level being searched is now zero and the maximum gray level being searched is now THRESH. The NEXT LOWER THRESH HI, shown at E, is the entropically selected threshold gray level for the gray level histogram corresponding to the gray level region between A and THRESH. The NEXT LOWER THRESH LO, shown at F, is the entropically selected threshold gray level for the gray level histogram corresponding to the gray level region between zero and A. The selection, searching and validating steps are then repeated recursively for the lower histogram using the next lower successive entropic threshold gray level, A, as the entropic threshold gray level.

[0093] The ANALYZE module as shown in Fig. 11 constitutes the core recursive kernel of the present invention and recursively partitions the histogram. The ANALYZE modules effectively zooms in on a specific region in gray level space to search for instances of the candidate object. The first step in the ANALYZE module as shown in Fig. 11 is to calculate the entropically selected threshold gray levels THRESH, THRESH HI AND THRESH LO as described above and as shown in block A of Fig. 11. As shown in block B, the module SEARCH IMAGE is run using the gray levels contained in the upper histogram. Decision diamond C then asks whether the number of new valid objects found is greater than the predetermined minimum number. If it is, then the module ANALYZE is run on the upper histogram recursively. If the number of valid objects found is not greater than the predetermined minimum number, then the module SEARCH IMAGE is run again using the gray levels contained in the lower histogram as shown in block E. Decision diamond F then asks whether the number of new valid objects found is greater than the predetermined minimum number. If it is, then ANALYZE is run on the lower histogram recursively as shown in block G. If it is not, then ANALYZE stops running, and the valid objects are returned as shown in block H of Fig. 11. With the method of the present invention, there is some latitude in selecting the range of values of the number of attributes to be checked for in the validation step during the recursive process.

[0094] In the preferred embodiment of the present invention, the searching step includes the sub-step of scanning a portion of the image for at least one candidate object using the entropically selected threshold gray level and tracing the candidate object having boundary gray levels determined by the entropically selected threshold gray level. The portion of the image scanned comprises a plurality of pixels, and each pixel has a gray level value less than the upper delimiter plus an increment. The upper delimiter also has the notation MAX. The increment is equal to the difference between MAX, the maximum gray level of the region being searched, and MIN, the minimum gray level of the region being searched, resulting in a new maximum gray level, Gray level_{\max} :

$$\text{Gray level}_{\max} = 2 \times \text{MAX} - \text{MIN} \quad (10)$$

Regions in the image where the gray level exceeds gray level_{\max} are ignored in the search.

[0095] The method of entropic thresholding used for identifying valid objects as described above is ideal in that it

uses the information contained in the image to optimally find all objects. However, this method does result in many objects being found that are not desired - i.e., artifacts. Often, the number of artifacts exceeds, by an order of magnitude, the number of desired objects. After performing the method of identifying valid objects as described above over time, one obtains information on the aggregate of images previously processed. Thus, a method which uses this information is particularly useful for an image on which one has no a priori information. With such a method, one can determine which conditions result in desired objects and which result in other non-desired objects, e.g., as in the optimization of conditions for identifying objects. The dual thresholding method as described herein uses entropic thresholding and generates a fixed threshold list from which to select a desired threshold pair based on the a priori information obtained from images previously processed.

[0096] Briefly, the dual thresholding method processes a set of images, keeping track of the thresholded pair used to find each object. This data is used to correlate the threshold pair used to obtain each object trace to the classification of objects as either desired or not desired objects. For each thresholding pair, the dual thresholding method determines if most of the objects found were of a desired or a not desired class. The dual thresholding method then constructs a list of threshold pairs that preferentially identify valid objects. This new threshold list is then used in the future for identifying valid objects and resolving redundancies of such objects in other images.

[0097] It may be desirable to fix a set of threshold pairs for the processing of the image to permit obtaining as many desired objects as possible, while as few non-desired objects as possible. Several methods may be used to do this, including: binary division of the gray level space; using a set of threshold pairs entropically determined from a representative image; or constructing a combined histogram of a set of images and entropically determining the set from that combined histogram. This latter approach is illustrated in Fig. 12.

[0098] Fig. 12 is a flow chart illustrating the steps of the dual thresholding method. A set of training images and a set of testing images is selected. As shown in block A of Fig. 12, a combined histogram for all training images is generated. The set of training images comprises a set of images representative of images for which object identification and categorization are desired. A set of testing images comprises images, again representative of images for which application of this method is desirable, but not part of the training image set. The valid objects in these images should be manually counted and marked by category to assist in judging the accuracy of the method. The combined histogram may then be processed as described above with respect to Figs. 10A - 10C. This shown in block B of Fig. 12, where a threshold pair list is then generated from the combined histogram, for example, using entropic thresholding. The training images are then processed - i.e., searched for candidate objects using the generated threshold pair list in order to obtain an object list as shown in block C. The list includes threshold pairs which generate an object trace. The objects are then matched to category marks, image by image, as shown in block D. Thus, each image has its own list as shown in block C; each object is matched in D. All object lists are then combined as shown in block E. The number of desired and not-desired objects found with each threshold pair is then counted, as also shown in block E. The desired threshold pairs are selected to create a reduced threshold pair list as shown in block F. As shown in block G, the reduced threshold pair list is then used to process test images in order to confirm the method performance, or new images for which automatic object classification is desired.

[0099] According to the method of the present invention, the validating step further includes the sub-step of checking for redundancies to prevent multiple identification of the valid object. Such redundancy checking is necessary since an object which has been recognized as valid after the current search may have been recognized as a valid object in an earlier search. In order to perform the redundancy checking sub-step, valid objects are classified as either homogeneous or inhomogeneous. Valid objects are also further classified as either a relatively large valid object or a small valid object. In addition, valid objects are classified as either contained in another valid object (interior valid objects), or never contained in another valid object (exterior valid objects).

[0100] According to the method of the present invention, the redundancy checking sub-step may be performed to delete inhomogeneous valid objects. When it is, the method of the present invention further includes the sub-step of deleting the large object when the large object contains more than one small object. Also, when the redundancy checking sub-step is performed to delete inhomogeneous valid objects, the method of the present invention also includes the sub-steps of calculating the average edge contrast of the large and the small valid objects and deleting the object having the smaller edge contrast when the large object contains only one small object. These sub-steps are performed by a module, CHK LIST, as shown in Figs. 13A - D for inhomogeneous valid objects.

[0101] As shown in block A of Fig. 13A, the first step of the CHK LIST module for deleting inhomogeneous objects is to define the previous count as the number of valid objects found prior to the current search. Then the tail object is defined as the initial candidate object found in the current search as shown in block B. The object count is initialized to one as shown in block C, and the head object is defined as the initial object in the total object list (i.e., the list of all objects found to date) as shown in block D. Decision diamond E asks whether the object count is greater than the previous count.

[0102] If the object count is greater than the previous count, CHK LIST advances to the first object in the total object list as shown in block A of Fig. 13B. Decision diamond B of Fig. 13B asks if CHK LIST is at the last object. If not, then

decision diamond C asks whether the valid object is contained within another valid object. If so, the object status is set to the status of the object within which it is contained as shown in block D, and CHK LIST advances to the next object as shown in block E. Also, if the object is not contained within another object, then CHK LIST advances to the next object as shown in block E. The loop through B - E continues until the next object of block E is the last object, at which point CHK LIST advances to the first object in the total object list as shown in block F. The object status attribute values for all the objects is set to "true" as shown in block G. "TRUE" in this context means valid, and "FALSE" means invalid. Decision diamond H then asks if CHK LIST is at the last object.

[0103] If it is, CHK LIST advances to the first object as shown in block A of Fig. 13C. Decision diamond B then asks again whether CHK LIST is at the last object. If it is, then the total number of objects is counted as shown in block C, and the difference between the total number of objects and the previous count is returned as shown in block D. If CHK LIST is not at the last object, decision diamond E asks whether the object status attribute value is false. If so, the object is deleted as shown in block F. If not, then CHK LIST advances the object as shown in block G, and CHK LIST asks again whether it is at the last object as shown in decision diamond B. The loop through B, E, F, and G continues until the advanced object of block G is the last object. At this point, the total number of objects is counted as shown in block C, and the difference between the total number of objects and the previous count is returned as shown in block D.

[0104] Returning to decision diamond H in Fig. 13B, if CHK LIST is not at the last object at this point, then it goes to decision diamond I, which asks whether the object contains more than one valid object. If so, then the object status attribute value is set to false as shown in block J, and CHK LIST advances to the next object as shown in block K. CHK LIST then returns to decision diamond H, asks whether it is at the last object and continues this process until the object does not contain more than one valid object. Then decision diamond A of Fig. 13D asks if the object is the only object contained within another object. If not, then CHK LIST advances to the next object as shown in block K of Fig. 13B, and the loop through H - K of Fig. 13B and A of Fig. 13D is repeated until the object is the only object contained within another object. If the object is the only object contained within another object, then decision diamond B asks whether the status attribute value of the object which contains the object is FALSE. If so, then CHK LIST advances to the next object as shown in block K of Fig. 13B, and the loop through H - K of Fig. 13B and A - B of Fig. 13D is repeated until the status attribute value of the object which contains the object is not FALSE. At this point, decision diamond N asks whether the edge contrast of the object which contains another object is greater than the edge contrast of the object. If so, then CHK LIST sets the object status attribute value to false as shown in block D, it advances to the next object as shown in block K in Fig. 13B, and the loop through H - K of Fig. 13B and A - C of Fig. 13D is repeated until the edge contrast of the object which contains another object is not greater than the edge contrast of the object contained in another object. Then CHK LIST sets the status of the object which contains the object to FALSE as shown in block E of Fig. 13D, and it advances to the next object as shown in block K of Fig. 13D until it is at the last object.

[0105] Returning to decision diamond E in Fig. 13A, if the object count is not greater than the previous count, then decision diamond F asks if the head object is contained within another object. If so, then the head object is advanced as shown in block G, and the object count is incremented as shown in block H. Decision diamond E again asks if the incremented object count is greater than the previous count. If so, CHK LIST advances to block A of Fig. 13B as explained above. If the incremented count is not greater than the previous count, the loop through F, G, H and E in Fig. 13A is repeated until the head object is not contained within another object. Then CHK LIST advances to decision diamond I of Fig. 13A, which asks if the tail object is the last object, or if the head object is contained within another object, then CHK LIST advances the head object as shown in block G, and the count is incremented as shown in block H. The loop through E, F, I, G and H is repeated until the tail object is not the last object or the head object is not contained within another object. Decision diamond J then asks whether the tail object is contained within another object. If it is, then the tail object is advanced as shown in block K of Fig. 13A, and the loop through I, J and K is repeated until the tail object is not contained within another object. Then CHK LIST goes to the module SET STAT as shown in Fig. 14 to set the status of the head and tail objects as shown in block L of Fig. 13A.

[0106] The redundancy checking sub-step further includes the sub-steps of comparing the areas of a plurality of valid objects and designating one of the valid objects as a large valid object and the other of the first and second valid objects as a small valid object and determining whether the small valid object is contained in the large valid object as defined by the four extremum points of the larger object for inhomogeneous objects. The module SET STAT as shown in Fig. 14 performs these sub-steps for inhomogeneous objects. The first step of SET STAT as shown in decision diamond A of Fig. 14 is to ask whether the head object is larger than the tail object. If so, then the head object is defined as the large valid object, and the tail object is defined as the small valid object as shown in block B. If the head object is not larger than the tail object, then the head object is defined as the small valid object, and the tail object is defined as the large valid object as shown in block C. Then decision diamond D asks whether the small object is contained within the large object. If not, then SET STAT is finished, as indicated by END oval E. If the small object is contained within the large object, then the large object type attribute value is set to a value indicating that it contains a small object as shown in block F. The type attribute value tells SET STAT whether an object is contained within another object

or whether the object contains another object. Also, the small object type attribute value is set to a value indicating that it is contained within a large object as shown in block G. Finally, the large object status attribute value is incremented as shown in block H. SET STAT is then finished, as indicated by the END oval I and returns to block L of Fig. 13A.

[0107] According to the method of the present invention, the redundancy checking sub-step may be performed to resolve redundancies in the homogeneous objects. When it is, the method of the present invention further includes the sub-steps of calculating the edge contrast of the large and small valid objects and deleting the large object where the average edge contrast of the large object is less than the average edge contrast of the small object and is less than a predetermined minimum edge contrast. The redundancy checking sub-step for resolving redundancies also includes the sub-steps of calculating the edge contrast of the large and small valid objects and deleting the small object where the average edge contrast of the large object is greater than the average edge contrast of the small object and is greater than the predetermined minimum contrast. These sub-steps are performed using the module CHK LIST for homogeneous objects as illustrated by the flow charts of Figs. 15A and 15B.

[0108] As shown in block A of Fig. 15A, the first step of the CHK LIST module, when run to delete homogenous objects, is to define the previous count as the number of valid objects found prior to the current search. Then the tail object is defined as the initial candidate object found in the current search as shown in block B. The object count is initialized to one as shown in block C, and the head object is defined as the initial object in the total object list as shown in block D. The object status attribute value is then set to true for all objects as shown in block E. Decision diamond F asks whether the object count is greater than the previous count.

[0109] If the object count is greater than the previous count, CHK LIST advances to the initial object in the total object list as shown in block A of Fig. 15B. Decision diamond B of Fig. 15B asks if CHK LIST is at the last object. If so, the total number of objects is counted as shown in block C, and the difference between the total number of objects and the previous count is returned as shown in block D. If CHK LIST is not at the last object, then decision diamond E asks whether the object status attribute value is FALSE. If so, the object is deleted as shown in block F. If the object status is not false, then object is advanced as shown in block G, and the CHK LIST module asks again whether it is at the last object as shown in decision diamond B. This process continues until CHK LIST reaches the last object, at which point the total number of objects is counted as shown in block C, and the difference between the total number of objects and the previous count is returned as shown in block D.

[0110] Returning to decision diamond F in Fig. 15A, if the object count is not greater than the previous count, then decision diamond G of Fig. 15A asks if the status attribute value of the head object is FALSE. If it is, then the head object is advanced as shown in block H, and the count is incremented as shown in block I. Decision diamond F then asks if the incremented object count is greater than the previous count. If so, CHK LIST advances to block A of Fig. 15B as explained above. The loop through G, H and I in Fig. 15A is repeated until the status of the object is not false. Then CHK LIST advances to decision diamond J of Fig. 15A, which asks if the tail object is not the last object and if the had object status attribute value is true. The answer to both these questions must be yes. If not, then CHK LIST advances the head object as shown in block H, and the count is incremented as shown in block I. The loop through F, G, H, I and J is repeated until the tail object is the last object and the head object status attribute value is true. Decision diamond K then asks whether the tail object status attribute value is true. If it is, then the edge status of the head and tail object is set as shown in block L of Fig. 15A and as shown in detail in Fig. 16 by a module SET STAT. CHK LIST then advances the tail object as shown in block M, and the loop through J, K, L and M is repeated. If the tail object status is not true, then CHK LIST advances the tail object as shown in block M, and the loop through J, K and M is repeated.

[0111] The module SET STAT as shown in Fig. 16 performs the sub-steps of comparing the areas of a plurality of valid objects and designating one of the valid objects as a large valid object and the other of the first and second valid objects as a small valid object and determining whether the small valid object is contained in the large valid object as defined by the four extremum points of the large object for homogeneous objects. As shown in decision diamond A of Fig. 16, the first step of SET STAT is to ask whether the head object is larger than the tail object. If so, the head object is defined as a large valid object, and the tail object is defined as the small valid object as shown in block B. If the head object is not larger than the tail object, then the head object is defined as the small valid object, and the tail object is defined as the large valid object. Decision diamond D of SET STAT then asks whether the small object is contained within the large object. If not, SET STAT stops running as shown by oval E. If the small object is contained within the large object, then decision diamond F asks whether the edge contrast of the large object is greater than the edge contrast of the small object, and whether the edge contrast of the large object is greater than the predetermined minimum edge contrast. If the answer to both of these questions is yes, then the large object status attribute value is set to true, and the small object status attribute value is set to false as indicated by block G, and the module stops running as indicated by oval H. If the answer to at least one of the questions in decision diamond F is no, then the small object status attribute value is set to true, the large object status attribute value is set to false as indicated by block I, and the module stops running as indicated by oval J.

[0112] The method of the present invention further includes the step of performing a final check for redundancies of

the valid object and resolving the redundancies to prevent multiple identification of the valid object. The final redundancy checking step further includes the sub-steps of comparing the areas of a plurality of valid objects and designating one of the valid objects as a large valid object and the other of the first and second valid objects as a small valid object and removing the large valid object when the small valid object and the large valid object overlap. The final redundancy checking step is performed by a module, FINAL CHK, as illustrated by the flow chart of Figs. 17A and 17B and a module INT STAT, as illustrated by the flow chart of Fig. 18. The modules FINAL CHK and INT STAT are the same for both homogeneous and inhomogeneous objects, and are thus only illustrated once.

[0113] The first step of FINAL CHK is to initialize the object attribute value to true for all objects as shown in block A of Fig. 17A. The counting index for counting valid objects is the initialized to one as shown in block B. The head object is defined as the initial object in a list of status attribute values as illustrated in block C. Decision diamond D then asks whether the counting index is less than the total number of objects. If not, the module FINAL CHK goes to block A of Fig. 17B.

[0114] As shown in block A of Fig. 17B, FINAL CHK advances to the first object. Decision diamond B asks whether FINAL CHK is at the last object. If it is not, then decision diamond C asks whether the object status attribute value is false. If not, then FINAL CHK advances to the next object as shown in block E, and decision diamond B again asks whether FINAL CHK is at the last object. The loop through B, C and E continues until FINAL CHK is at the next object. If the object status attribute value is false, then the object is deleted as shown in block D. FINAL CHK then advances to the next object as shown in block E, and decision diamond B asks whether the FINAL CHK at the last object. The loop through B - E continues until the next object is the last object, at which point FINAL CHK advances to the first object as shown in block F. The count is then initialized to one as shown in block G. Decision diamond H then asks whether FINAL CHK is at the last object. If it is not, then the count is incremented as shown in block I, and FINAL CHK advances to the next object is shown in block J. Decision diamond H again asks whether FINAL CHK is the last object, and the loop through H, I and J continues until FINAL CHK is at the last object. Then the total number of valid objects as contained in count is returned as shown by block K of Fig. 17B.

[0115] Returning to decision diamond D of Fig. 17A, if the counting index is less than the total number of objects, then the tail object is defined as the next object beyond the head object as shown in block E. Decision diamond F then asks if the status attribute value of the head object is true. If not, the FINAL CHK advances the head object as shown in block G and increments the counting index as shown in block H. FINAL CHK then returns to decision diamond D, and the loop through D - I continues until the status attribute value of the head object is true. Then decision diamond I asks whether the tail object is not the last object and whether the head object status attribute value is true. If at least one of these conditions is not met, then FINAL CHK advances the head object as shown in block G and increments the index as shown in block H. FINAL CHK then returns to decision diamond D, and the loop through D - I continues until the answer to both questions in decision diamond I is yes. Then decision diamond J asks whether the tail object status attribute value is true. If not, FINAL CHK advances the tail object as shown in block L of Fig. 15A, and the loop through I, J and L is repeated until the tail object status attribute value is true. Then FINAL CHK runs a module INT STAT, as shown in block K of Fig. 17A and advances the tail object as shown in block L.

[0116] The steps of the module INT STAT as illustrated in block K of Fig. 17A are shown in detail in Fig. 18. Decision diamond A of Fig. 18 asks whether the head object is larger than the tail object. If so, the head object is defined as the large valid object, and the tail object is defined as the small valid object as shown in block B. If the head object is not larger than the tail object, then the head object is defined as the small valid object, and the tail object is defined as the large valid object as shown in block C. Decision diamond D then asks whether the small valid object is contained in the large valid object. If not, then INT STAT is at its end, as shown by oval E. If the small valid object is contained in the large valid object, then the large object status attribute value is set to false as shown in block F, and INT STAT is at its end as shown by oval G.

[0117] The method of the present invention further includes the step of filtering the image. The filtering step is performed either after the method of the present invention is used as a screening process or as a recursive process. The filtering step is performed by a module, CALCON, as shown in general in Figs. 19A and in detail in Figs. 19B - D for inhomogeneous valid objects and in Figs. 19A and 19B for homogeneous valid objects.

[0118] The filtering step for filtering inhomogeneous objects comprises three prongs defining certain conditions under which inhomogeneous objects should be deleted. The first prong in the filtering step deletes inhomogeneous objects using a heavy filter when the number of inhomogeneous valid objects retained after the final redundancy check as performed by FINAL CHK as described above is less than or equal to a predetermined minimum number. The second prong deletes inhomogeneous valid objects using a medium filter when the number of retained inhomogeneous valid objects is greater than the predetermined minimum number and less than a predetermined maximum number. The third prong deletes inhomogeneous valid objects using a light filter when the number of retained inhomogeneous valid objects is greater than or equal to the predetermined maximum number. Preferably, the three prongs of the filtering step for deleting inhomogeneous valid objects are repeated twice immediately after each respective prong. It is also preferable that the prongs are repeated after the recursive partitioning of the histogram when the method of the present

invention is used as a recursive process. However, the prongs may also be repeated after the validation step when the method of the present invention is used as a screening process.

[0119] As shown in block A of Fig. 19A, the first step of CALCON for deleting inhomogeneous objects is to advance to the initial object. Decision diamond B then asks whether the number of valid objects is greater than the predetermined minimum number. If not, then a heavy filter is applied as shown in block C. If the number of valid objects is greater than the predetermined minimum number, then decision diamond D asks whether the number of valid objects is less than a predetermined maximum number. If it is, then a medium filter is applied as shown in block E. If it is not, then a light filter is applied as shown in block F. Decision diamond G then asks whether CALCON is at the last object. If it is, then the module stops running as shown at oval H. If it is not, then decision diamond I asks whether the object is within a predetermined minimum pixel distance from the edge of the image boundary and whether the area is less than a predetermined minimum area. If the answer to both these questions is yes, the object is deleted as shown in block J of Fig. 19A, and the object is advanced to the next object as shown in block M. Decision diamond G then asks whether CALCON is at the last object. The loop through G - M continues until the answer to at least one of the questions in I is no. If the answer to at least one of these questions is no, then decision diamond K asks whether the filter is light. If so, then a light filter is applied as shown in block L and in detail in Fig. 19B. CALCON then advances to the next object as shown in block M, and the loop through G, I, J, K, L and M continues until the filter is not light. Then decision diamond N asks whether the filter is medium. If so, then a medium filter is applied as shown in block O of Fig. 19A and in detail in Fig. 19C. CALCON then advances to the next object as shown in block M, and the loop through G, I, J, K, N, O and M continues until the filter is not medium. A heavy filter is then applied as shown in block P, and in detail in Fig. 19D. CALCON then advances to the next object as shown in block Q. The loop through G, I, J, K, N, P and Q continues until CALCON is at the last object as asked in decision diamond G. At this point, the module stops running as shown by oval H.

[0120] An inhomogeneous object is deleted using a light filter when the inhomogeneous object meets certain criteria as illustrated in Fig. 19B. Decision diamond A asks whether the object was ever contained in another object. If so, the module stops running as shown at END oval B. If the object was never contained in another object, then decision diamond C asks whether the inhomogeneous object area is less than the predetermined minimum area. If so, the object is deleted as shown by block D, and the module stops running as indicated by END oval E. If the inhomogeneous object area is not less than the predetermined minimum area, then decision diamond F asks whether the object edge contrast is less than the predetermined minimum edge contrast and whether the center of mass gray level is less than the entropic gray level for the lower histogram as defined by the recursive process. If the answer to both these questions is yes, the object is deleted as shown by block G. If the answer to at least one of the questions in diamond G is no, then the module stops running as shown by END oval I.

[0121] An inhomogeneous object is deleted using a medium filter when the inhomogeneous object meets certain criteria as illustrated in Fig. 19C. Decision diamond A of Fig. 19C asks whether the object was ever contained within another object. If not, then decision diamond B asks whether the inhomogeneous valid object area is less than the predetermined minimum area. If so, the object is deleted as shown in block C, and the module stops running as indicated by END oval D. If the object area is not less than the inhomogeneous object minimum area, then decision diamond E asks whether the object edge contrast is less than the predetermined minimum edge contrast and whether the center of mass gray level is less than the original entropic threshold gray level. If the answer to both these questions is yes, then the object is deleted as shown in block F, and the module stops running as indicated by the END oval G. If the answer to at least one of these questions is no, the module stops running as indicated by the END oval H. Returning to decision diamond A, if the object was previously contained in another object, then decision diamond I asks whether the inhomogeneous object edge contrast is less than the predetermined minimum edge contrast, whether the inhomogeneous object center of mass gray level is less than the original entropic threshold gray level and whether the inhomogeneous object area is less than the predetermined minimum area. If the answer to all these questions is yes, the object is deleted as shown in block J and the module stops running as shown by END oval K. If the answer to at least one of these questions is no, the module stops running as shown by END oval L.

[0122] An inhomogeneous object is deleted using a heavy filter when the inhomogeneous object meets certain criteria as illustrated in Fig. 19D. Decision diamond A of Fig. 19D asks whether the object was ever contained within another object. If it was not, then decision diamond B asks whether the inhomogeneous valid object has an area less than a predetermined minimum area. If the answer to this question is yes, the object is deleted as shown in block D. If not, then decision diamond C asks whether the inhomogeneous valid object has an edge contrast greater than a predetermined minimum edge contrast and an area greater than a predetermined minimum area and less than a predetermined maximum area. If so, the object is deleted as shown in block D, and the module stops running. If not, then the module stops running, as indicated by the END oval F. If the inhomogeneous valid object was previously contained within another object, then decision diamond G asks whether the inhomogeneous object center of mass gray level is less than the original entropic threshold gray level and whether the area greater than the predetermined minimum area and less than the predetermined maximum area. If the answer to both these questions is yes, the object is deleted as shown

in block H, and the module stops running as indicated by END oval I. If the answer to at least one of these questions is no, then decision diamond J asks whether the inhomogeneous object center of mass gray level is less than the entropic threshold gray level for the lower histogram as defined by the recursive process. If so, the object is deleted as shown in block K, and the module stops running as indicated by the END oval L. If not, then the module stops running as indicated by END oval M.

[0123] The steps for running CALCON to filter homogeneous objects are shown in the flow charts of Figs. 20A and 20B. As shown in block A of Fig. 20A, the first step of CALCON for filtering homogeneous objects is to advance to the first object. Decision diamond B then asks whether CALCON is at the last object. If so, CALCON is finished as indicated by END oval C. If not, decision diamond D asks whether the homogeneous valid object is within a predetermined minimum pixel distance from the edge of the image boundary and whether the homogeneous valid object has an area less than the predetermined minimum area. If the answer to both these questions is yes, then the homogeneous object is deleted as shown in block E, and CALCON advances the object as shown in block F. Decision diamond B asks whether the advanced object is the next object, and the loop through B - F continues until at least one of the conditions in decision diamond D is not met. Then decision diamond G asks whether the homogeneous object area is less than the predetermined minimum area. If it is, then the object is deleted as shown in block E, and the loop through B - G continues until the homogenous object area is not less the predetermined minimum area. Then, decision diamond H asks whether the homogenous object has an edge contrast less than the predetermined minimum edge contrast and whether the homogeneous object area is less than the predetermined minimum area. If the answer to both these questions is yes, the object is deleted as shown in block E, and CALCON advances the object as shown in block F. The loop through B - H continues until the answer to one of the questions in decision diamond H is no.

[0124] Then decision diamond I of Fig. 20B asks whether the homogeneous object center of mass gray level is less than the original entropic threshold gray level and whether the edge contrast of the homogeneous object is less than the predetermined minimum edge contrast. If the answer to both these questions is yes, then the object is deleted as shown in block J. CALCON then advances the object as shown in block K, and the loop through B - I continues until the answer to at least one of the questions in decision diamond I is no. Then decision diamond L asks whether the shape factor of the homogenous object is less than a predetermined shape factor and whether the edge contrast of the homogeneous object is less than the predetermined minimum edge contrast. If the answer to both these questions is yes, then the object is deleted as shown in block J. The loop through B - L continues until the answer to at least one of the questions in decision diamond L is no. Then decision diamond M asks whether the edge contrast of the homogeneous object is less than an absolute predetermined minimum edge contrast. If it is, then the object is deleted as shown in block J, and CALCON advances the object. The loop through B - M continues until the object edge contrast is not less than the absolute predetermined minimum edge contrast. At this point, CALCON advances the object until it at the last object, and the module stops running as shown by END oval C.

[0125] The image examined by the method of the present invention comprises at least one isolated valid object and at least one candidate clump of valid objects. An isolated valid object is identified by its degree of circularity. The candidate clump of valid objects has an area greater than the mean area of the isolated valid objects and a shape factor less than a predetermined clump shape factor. The method of the present invention further includes the steps of identifying the candidate clump of inhomogeneous valid objects and determining the number of inhomogeneous valid objects in the candidate clump. Unclumping of homogeneous valid objects is inherent in the recursive process of the first embodiment of the present invention.

[0126] A module, UNCLUMP, is used to identify the candidate clump and determine the number of valid objects in the candidate clump. The steps of UNCLUMP are shown in the flow charts of Figs. 21A and 21B and Figs. 22A and 22B. As shown in block A of Fig. 21A, the first step of UNCLUMP is to advance to the initial object. Decision diamond B then asks whether UNCLUMP is at the last object. If not, then decision diamond C asks whether the object shape factor is greater than the predetermined clump shape factor. If it is, then the object is added to the list of isolated objects as shown in block D, and the count is incremented as shown in block E. UNCLUMP advances to the next object as shown in block F. UNCLUMP then returns to decision diamond B, and the loop through B - F continues until the object shape factor is not greater than the predetermined clump shape factor. UNCLUMP then advances to the next object as shown in block F. UNCLUMP then returns to decision diamond B, and the loop through B, C and F continues until UNCLUMP is at the last object.

[0127] When UNCLUMP reaches the last object, the mean object area of the isolated objects is calculated as shown in block G of Fig. 21A. UNCLUMP then advances to the first object as shown in block H, and decision diamond I asks whether UNCLUMP is at the last object. If it is not, then decision diamond J asks whether the object area is greater than the mean object area and whether the object shape factor is less than the predetermined clump shape factor. If the answer to both these questions is yes, then the object status attribute value is set to true as shown in block K, and UNCLUMP advances to the next object as shown in block L. The loop through I - L continues until UNCLUMP is at the last object. UNCLUMP then advances the first object as shown in block M, and the object counter is set to zero as shown in block N of Fig. 21A.

[0128] UNCLUMP then goes to decision diamond O as shown in Fig. 21B, which asks whether it is at the last object. If it is, then the number of objects found is returned as shown in block P. If UNCLUMP is not at the last object, then decision diamond Q asks whether the object status attribute value is true. If the value is true, then UNCLUMP processes the clump by incorporating a module, PROCESS CLUMP, and returns the number of objects in the clump as shown in block R. The steps of PROCESS CLUMP are illustrated in the flow chart of Figs. 22A and 22B. The count is incremented by the number of objects in the clump as shown in block S. If the object status attribute value is not true, then the isolated object is displayed as shown by block T. The count is incremented as shown by block U, and UNCLUMP advances to the next object as shown by block V. UNCLUMP then returns to decision diamond O. The loop through O - V continues until UNCLUMP is at the last object, at which point the number of objects found is returned as shown in block P.

[0129] The first step in PROCESS CLUMP is to trace the boundary of the clump as shown in block A of Fig. 22A. An outline of a candidate clump is illustrated in Fig. 21A. Perimeter points are shown at A, B, C and D, where A and C are spaced a maximum distance, d, from the center of mass of the clump. A distance buffer as shown in Fig. 22B is then created. The distance buffer comprises a plurality of distance values representing the distance of each perimeter point of the candidate clump to the center of mass of the candidate clump. The distance buffer includes a plurality of valleys and peaks as shown in Fig. 23B. The distance buffer is searched for a first valley, V_1 , as shown in block C of Fig. 22A using a hysteresis-based, valley-finding algorithm.

[0130] A hysteresis-based, peak-finding algorithm is an algorithm known to one skilled in the art to identify the peak values in a set of data where there may be fluctuations in the values of the data points. To ensure that spurious peaks are not found in the data set, a hysteresis "window" is defined with a certain width of values. For a peak to be counted as a true peak, two conditions must be satisfied: (a) the value of the peak data point must be at least one window width above values of points prior to the peak; and (b) the value of data points subsequent to the peak must fall at least one window width below the peak value before they start to rise. This ensures that small ripples in the data with amplitudes less than the window width are not included as peaks. A hysteresis-based, valley-finding algorithm is the same as the hysteresis-based, peak-finding algorithm, with the only difference being that the data set is initially inverted prior to analysis. Inverting the data set converts the original valleys into peaks, and the original peaks into valleys. This allows one to use the hysteresis-based, peak-finding algorithm described above to find the original valleys. In the present invention, the window width was scaled to 10% of the difference in values between the maximum data point and the minimum data point. For data where this difference was very small, the window width was scaled to 75% of the difference.

[0131] Decision diamond D as shown in Fig. 22A asks whether the first valley point is the last point in the distance buffer. If not, the distance buffer is shifted by a distance, D, as shown by block E of Fig. 22A. Distance D is shown in Figs. 23B and 23C and is equal to the distance to the first valley from a first distance value as defined by a first predetermined point on the perimeter of the candidate clump. The values at the start of the distance buffer are wrapped around to the end of the buffer. The number of peaks in the shifted buffer is then counted as using the hysteresis-based, peak-finding algorithm as shown in block F to calculate a first value, N_1 , for the number of valid objects in the clump, and PROCESS CLUMP advances to block H of Fig. 22B. First value, N_1 , is calculated according to the following formula:

$$N_1 = (\# \text{ of peaks}/2) + 1 \quad (11)$$

The number of peaks in the distance buffer is divided by two to obtain a result which comprises an integer and a fractional value. The result is rounded up to the next higher integer and one is added to the next higher integer. If the first valley point is the last point in the distance buffer, then first value N_1 is set to one as shown in block G of Fig. 22A, and PROCESS CLUMP advances to block H of Fig. 22B.

[0132] The number of valid objects in a candidate clump is also calculated by a second method which uses the mean area of the isolated valid objects. This step is shown in block H of Fig. 22B. According to the second method, the area of the candidate clump is calculated. The area of the candidate clump is then divided by the mean area of the isolated valid object to calculate a second value, N_2 , for the number of valid objects in the clump. The ratio N_2 determines the number of isolated colonies which can "fit in" the clump.

$$N_2 = \frac{\text{Area of clump}}{\text{Mean area of isolated valid object}} \quad (12)$$

[0133] To account for backgrounds where the valid object distribution for isolated valid objects can vary greatly, the number of valid objects in a clump is calculated according to a third method. This step is shown in block I of the flow

chart of Fig. 22B. According to the third method, the area of a candidate clump is calculated. The area is then divided by the mean area of the isolated valid objects minus one-half the standard deviation of the mean area of the isolated valid object to give a third value, N_3 , for the number of valid objects in the clump.

5

$$N_3 = \frac{\text{Area of clump}}{\text{Mean area isolated object} - 0.5 (\text{std. dev. of area of isolated object})}$$

[0134] The average value of second and third values, N_2 and N_3 is then computed. The average value is compared to first value N_1 as shown in decision diamond J. If first value N_1 is greater than three, or if the average of N_2 and N_3 is greater than N_1 , the average value is selected as the number of valid objects in the candidate clump as shown by block K. Otherwise, the first value is selected as the number of valid objects in the candidate clump as shown by block L.

[0135] Decision diamond M of Fig. 22B asks whether PROCESS CLUMP is using first value N_1 as an estimate of the number of valid objects in the clump. If it is not, then the clump is traced in a singular color indicative of the number of valid objects in the candidate clump as shown in block O. PROCESS CLUMP then returns the number of valid objects in the clump as shown in block P. If PROCESS CLUMP uses first value N_1 as an estimate of the number of valid objects in the candidate clump, then the clump is traced in a plurality of colors as shown in block N of Fig. 22B. The colors used for tracing are indicative of the number of valid objects in the clump, where all the points of the perimeter of the clump between the adjacent valleys on the distance buffer are traced in the same color. UNCLUMP then returns the number of valid objects in the clump as shown in block P.

[0136] As an alternative to the method described above, any one of the three methods may be used individually to determine the number of valid objects in a clump.

[0137] The present invention further includes a first, second and third method of determining the number of clumped homogeneous valid objects in a background comprising at least one clump of homogeneous valid objects and at least one isolated homogeneous valid object, where the isolated valid object is identified by its degree of circularity. According to the three methods, at least one candidate clump is identified, where the candidate clump has an area greater than the mean area of the isolated valid objects and a shape factor less than a predetermined clump shape factor. The steps for determining the number of clumped homogeneous valid objects in a background are the same as those shown in Figs. 21A and 21B and 22A and 22B.

[0138] Further in accordance with the first method of determining the number of clumped homogeneous valid objects of the present invention, a distance buffer is created as discussed above. The distance buffer comprises a plurality of distance values representing the distance of each perimeter point of the candidate clump to the center of mass of the candidate clump. The distance buffer includes a plurality of valleys and peaks as shown in Fig. 23B and 23C. The distance buffer is searched for a first valley, V_1 , using a hysteresis-based, valley-finding algorithm as described above. The distance buffer is then shifted by a distance, D , equal to the distance to the first valley from a first distance value as defined by a first predetermined point on the perimeter of the candidate clump. The distance values at the start of the buffer are wrapped around to the end of the buffer. As in the above-described method, the number of peaks is then counted in the shifted buffer using a hysteresis-based, peak-finding algorithm to calculate a value, N_1 , for the number of valid objects in the clump. The first value, N_1 , is calculated according to equation (11) as set forth above.

[0139] In accordance with the second method of determining the number of clumped valid objects in a background, the number of valid objects is also calculated using the mean area of the isolated valid objects. In accordance with the second method, the area of the candidate clump is calculated. The area of the candidate clump is divided by the mean area of the isolated valid object to calculate a value, N_2 , for the number of valid objects in the clump, where N_2 is given by equation (12) as set forth above.

[0140] In accordance with the third method of determining the number of clumped valid objects in a background, the number of valid objects is also calculated using the mean area of the isolated valid objects. With this method, the area of the candidate clump is calculated. The area is then divided by the mean area of the isolated valid objects minus one-half the standard deviation of the mean area of the isolated valid object to calculate a value, N_3 , for the number of valid objects in the clump as set forth in equation (13) above.

[0141] In accordance with the first embodiment of the present invention, there is provided an image analysis system for identifying at least one valid object in a background. The valid object has at least one predetermined attribute value which represents the definition of a valid object of an object to be identified. A block diagram of the system of the present invention is shown in Fig. 24. A system for identifying at least one valid object in a background is shown generally at 10 in Fig. 24.

[0142] The system of the present invention comprises means for generating an image of the object and the background. As shown in Fig. 24, the means for generating an image of the object and the background comprises a camera 12. Although a CCD camera is generally used with the present invention, any type of camera may be used without departing from the general principles of the present invention.

[0143] The system of the present invention also comprises means for digitizing and storing the image. The means

for digitizing and storing the image comprises a frame grabber 14 as shown in Fig. 24. The frame grabber digitizes and stores the video image in one frame, as known to one skilled in the image processing art. Alternatively, the means for digitizing and storing the image comprises a video digitizer, which also digitizes and stores the image, although not necessarily in one frame. The system of the present invention further comprises means for displaying the image. The means for displaying the image comprises a monitor 16 as shown in Fig. 24.

[0144] The system of the present invention also comprises computer means. The computer means comprises a computer system 18 as shown in Fig. 24. The computer system comprises a central processing unit (CPU) and a memory. The computer means also includes a driver 20, a kernel 22 and a post-scan filter 26 as shown in Fig. 24. Driver 20 stores the definition of the valid object. The entropic kernel 22 generates a gray level histogram of the image and entropically selects a threshold gray level such that the entropy function of the histogram is maximized. Entropic kernel 22 also searches the image for at least one candidate object and validates the candidate object having the valid object predetermined attribute value to identify the valid object. The validated objects are represented by box 24 in Fig. 24. The driver and the kernel may comprise software incorporated in the memory. Alternatively, the driver and the kernel may be programmed into a programmable, read-only memory (PROM) from which the software may be retrieved. The post-scan filter is shown at 26 in Fig. 24 and provides a final check to remove redundancies in overlapping objects as described above.

[0145] According to a second embodiment of the present invention, there is provided another method of identifying at least one valid object having at least one predetermined attribute in a background. Fig. 25 is a block diagram showing the overall method of the second embodiment of the present invention.

[0146] The method comprises the steps of generating an image of the object and the background. An image is generated as shown in block A of Fig. 25. As in the first embodiment, the image of the object and the background may be generated by a camera. The image is then digitized and stored by a frame grabber or a video digitizer.

[0147] The method also comprises the step of generating a gray level co-occurrence matrix of the image. This step is shown in block B of Fig. 25. A co-occurrence matrix is shown in Fig. 28. The co-occurrence matrix is generated in accordance with known principles as set forth in the Pal and Pal article entitled "Entropic Thresholding" cited above. The two-dimensional co-occurrence matrix suggested by Pal and Pal includes, in principle, both the first order and second order gray level statistics which are important to the visual process. The matrix elements p_{ij} of the co-occurrence matrix as described by Pal and Pal represent the probability of co-occurrence of gray levels i and j spaced apart by a given distance parameter d . The distance parameter is assumed to be equal to one in considering the "nearest neighbor" texture. The probability of co-occurrence of two gray levels is the result of averaging the probability over four directions (horizontal, vertical, right diagonal and left diagonal). The co-occurrence matrix is made asymmetrical by searching the image only one way in each of the four directions. For example, in the case of horizontal searching, each row might be searched from left to right.

[0148] The present invention uses four modules, HORIZ, VERT, LFT DIAG and RT DIAG, to generate the gray level co-occurrence matrix of the region of interest of the image. The steps for generating the gray level matrix are shown in the flow chart of Fig. 26. As shown in block A of Fig. 26, HORIZ, VERT, LFT DIAG and RT DIAG first calculate a matrix of the region of interest of the image. They then calculate the values to be used subsequently in the calculation of the entropy function, H_i , for each gray level, i , as shown in block B of Fig. 24 for each gray level of the matrix. The results of this calculation are stored in a global look-up table as shown in block C.

[0149] The method of the present invention also includes the step of entropically selecting a threshold gray level such that the entropy function of the matrix is maximized. This step is shown generally in block C of Fig. 25 and is performed by the ENTROPY module as shown in Fig. 27. As shown in block A of Fig. 27, the first step in maximizing the entropy function of the matrix is to initialize the maximum entropy function to a minimum value. The partitioned threshold gray level is then initialized to a minimum value as shown in block B of Fig. 27.

[0150] The step of entropically selecting a threshold gray level also includes the sub-step of sequentially partitioning the matrix at each gray level into a first, second, third and fourth quadrant. This sub-step partitions the matrix as shown in Fig. 28 into a first quadrant, A, a second quadrant, B, a third quadrant, C, and a fourth quadrant, D. In this partition, the "diagonal" quadrants A and C describe the internal texture of the background and the object, respectively. The "off-diagonal" quadrants B and D describe the edge texture between the object and the background.

[0151] Partitioning a co-occurrence matrix to entropically select a maximum threshold gray level was developed by Pal and Pal as described in "Entropic Thresholding", cited above. Pal and Pal suggest two entropy maximization schemes in their article. In the first scheme, the diagonal entropy function is maximized, and in the second case, the off-diagonal entropy function is maximized. Maximizing the diagonal entropy is analogous to maximizing the entropy of the gray level histogram, since it really involves utilizing gray level information from the interiors of the object and the background. The off-diagonal terms relate to edge information, which is not present in the gray level histogram. The description given below relates to off-diagonal entropy, but may be extended to diagonal entropy without any loss of generality. The off-diagonal entropy can be defined as:

$$H_t^{\text{off}} = (H(O/B) + H(B/O))/2 \quad (14)$$

where

$$H(O/B) = - \sum_{i=1}^s \sum_{j=s+1}^{N_{\text{gray}}} p_{ij}^B \ln p_{ij}^B \quad (15)$$

$$H(B/O) = - \sum_{i=s+1}^{N_{\text{gray}}} \sum_{j=1}^s p_{ij}^D \ln p_{ij}^D \quad (16)$$

$$p_{ij}^B = \frac{p_{ij}}{p_B}, p_{ij}^D = \frac{p_{ij}}{p_D} \quad (17)$$

and

$$p_B = \sum_{i=1}^s \sum_{j=s+1}^{N_{\text{gray}}} p_{ij}, p_D = \sum_{i=s+1}^{N_{\text{gray}}} \sum_{j=1}^s p_{ij} \quad (18)$$

The basic parameter P_{ij} is defined by:

$$p_{ij} = \frac{t_{ij}}{\sum_i \sum_j t_{ij}} \quad (\text{summed from } i, j = 1, \dots, N_{\text{gray}}) \quad (19)$$

where t_{ij} is the frequency of co-occurrence of the gray level pair (i, j) . After substituting in for P_{ij} , and performing the requisite algebra, one obtains:

$$H_T^{\text{off}} = \ln \left(\sum_{i=1}^s \sum_{j=s+1}^{N_{\text{gray}}} t_{ij} \right) + \ln \left(\sum_{i=s+1}^{N_{\text{gray}}} \sum_{j=1}^s t_{ij} \right) - \frac{\sum_{i=1}^s \sum_{j=s+1}^{N_{\text{gray}}} t_{ij} \ln t_{ij}}{\sum_{i=1}^s \sum_{j=s+1}^{N_{\text{gray}}} t_{ij}} - \frac{\sum_{i=s+1}^{N_{\text{gray}}} \sum_{j=1}^s t_{ij} \ln t_{ij}}{\sum_{i=s+1}^{N_{\text{gray}}} \sum_{j=1}^s t_{ij}} \quad (20)$$

[0152] According to the present invention, the entropy function of the matrix is then computed. This step is shown in block C of Fig. 27. The entropy function of the matrix is defined as the sum of the entropy functions of the second and third, or "off-diagonal" quadrants. Alternatively, the entropy function of the matrix is defined as the sum of the entropy functions of the first and fourth, or "diagonal" quadrants, of the matrix. A threshold gray level is selected such that the entropy function of the matrix is maximized. Decision diamond D of Fig. 27 asks whether the entropy function of the matrix is greater than the maximum entropy function as initialized in block A. If it is, then the maximum entropic

threshold gray level is updated to the partitioned threshold gray level as shown in block E of Fig. 27. The maximum entropy function is then updated to the entropy function of the matrix as shown in block F of Fig. 27. If the entropy function of the matrix is not greater than the maximum entropy function, then decision diamond G of the ENTROPY module of Fig. 27 asks whether the partitioned threshold gray level equals the maximum threshold gray level. If so, the maximum entropic threshold gray level is returned as shown in block H of Fig. 27. If not, then the partitioned threshold gray level is incremented as illustrated in block I of Fig. 27, and the incremented partitioned threshold gray level is returned to block C, where the entropy function of the incremented, partitioned threshold gray level is computed. This process is repeated until the partitioned threshold gray level equals the maximum threshold gray level as illustrated in block G. The maximum entropic threshold gray level is then returned as shown in block H. The maximum entropic threshold gray level divides the original matrix into an upper and lower matrix corresponding to quadrants B and D, respectively, for the off-diagonal case.

[0153] The method of the second embodiment of the present invention also includes the step of searching the image for at least one candidate object, wherein the candidate object has at least one candidate object attribute value. This step is shown in block D of Fig. 25. The searching step includes the sub-steps of scanning the image for at least one candidate object using the entropically selected threshold gray level and tracing the candidate object having boundary gray levels determined by the entropically selected threshold gray level. The searching step is illustrated by the flow chart SEARCH IMAGE, which is the same for the co-occurrence matrix as the flow chart shown in Fig. 6 for the histogram. In addition, the flow charts FIND OBJECT of Fig. 7 and TRACE OBJECT of Fig. 8 for the histogram of the first embodiment are the same for the co-occurrence matrix of the second embodiment. The method of the second embodiment further comprises the step of validating the candidate objects having the valid object predetermined attribute values, thereby identifying the valid object. This step is shown in block E of Fig. 25 and is also performed by TRACE OBJECT.

[0154] The method as described thus far can be used as a screening process to identify at least one valid object. As in the first embodiment, for a more stringent identification process, it is possible to recursively search the gray level space by using the co-occurrence matrix. The steps of the recursive process using a co-occurrence matrix are performed by the module ANALYZE and are shown in the flow chart of Fig. 29.

[0155] As shown in block A of Fig. 29, the first step of ANALYZE for a co-occurrence matrix is to calculate the entropically selected threshold gray levels, THRESH, THRESH HI and THRESH LO. ANALYZE then runs SEARCH IMAGE using the gray levels contained in the upper matrix as shown in block B. Decision diamond C then asks whether the number of new valid objects found by SEARCH IMAGE is greater than a predetermined minimum number. If so, then ANALYZE is run on the upper matrix recursively as shown in block D. The recursion continues until the number of new valid object found by SEARCH IMAGE is not greater than the predetermined minimum number. Then, SEARCH IMAGE is run using the gray levels contained in the lower matrix as shown in block E. Decision diamond F then asks whether the number of new valid objects found is greater than the predetermined minimum number. If so, then ANALYZE is run on the lower matrix recursively as shown in block G, and the recursion continues until the number of new valid objects found is not greater than the predetermined minimum number. At this point, ANALYZE returns the valid objects as shown in block H.

[0156] The remaining steps of the method of the first embodiment and the system as described above with respect to Fig. 24 for the first embodiment are equally applicable to the second embodiment. Thus, the modules as shown in Figs. 9 and 12-22 may be run for the co-occurrence matrix embodiment without any loss of generality.

[0157] In accordance with a third, or iterative embodiment of the present invention, also referred to as IETNA, (iterative entropic thresholding in natural analysis) there is provided a method of identifying at least one valid object having at least one predetermined attribute value in a background. By iterative, it is meant the process of dividing a histogram into upper and lower histograms for a predetermined number of N divisions or iterations so as to create N + 1 sub-histograms. A block diagram of the method according to the third embodiment of the present invention is shown in Fig. 30.

[0158] The method according to the third embodiment of the present invention comprises the step of generating an image of the object and the background. An image is generated as shown in block A of Fig. 30. As in the first and second embodiments, the image of the object and the background may be generated by a camera. The image is then digitized and stored by a frame grabber or a video digitizer.

[0159] The method of the present invention also comprises the step of generating a gray level histogram of the image. This step is shown in block B of Fig. 30. The steps of the module for generating gray level histogram for the third embodiment are the same as those shown in the flow chart of Fig. 3.

[0160] The method of the third embodiment of the present invention also includes the step of selecting N global entropic threshold gray levels as shown in block C of Fig. 30 and subdividing the gray level histogram into N + 1 sub-histograms using each of the entropically selected threshold gray levels as shown in block D. To illustrate these steps, Fig. 31 shows a gray level histogram having N global entropic threshold gray levels, $t_1 - t_N$. The gray level histogram is divided into N + 1 sub-histograms using each of the global entropic threshold gray levels. The steps for selecting N

global entropic threshold gray levels are performed by the module ENTROPY and are the same as those shown in the flow chart of Fig. 4.

[0161] The selecting step includes the sub-steps of sequentially partitioning the histogram at each gray level into a first and a second partition. The entropy function is then computed for each partition. The entropy function of the histogram is defined as the sum of the entropy functions of the first and second partitions. A global entropic threshold gray level is then selected such that the entropy function of the histogram is maximized. The gray level histogram is then subdivided using the global entropically selected threshold gray level as defined above which maximizes the entropy function of the histogram as an upper delimiter and a lower delimiter to create an upper histogram and a lower histogram. The partitioning, computing and selecting steps are then repeated, where the repetition of the selecting step selects a next successive global entropic threshold gray level. The subdividing step is then repeated using the next successive entropic threshold gray level as the global entropic threshold gray level as defined in the selecting step to iteratively calculate the N global entropic threshold gray levels.

[0162] The iterative method of the present invention further includes the step of searching portions of the image corresponding to each sub-histogram using each entropically selected threshold gray level for at least one candidate object. The candidate object has at least one candidate object attribute value. The step of searching portions of the image is shown in block E of Fig. 30. The searching step includes the sub-steps of scanning the image for at least one candidate object using each global entropically selected threshold gray level and tracing the candidate object having boundary gray levels determined by each global entropically selected threshold gray level. The step of searching portions of the image is performed by the modules SEARCH IMAGE, FIND OBJECT AND TRACE OBJECT as described above for the first embodiment and as shown in Figs. 6 - 8. In addition, the module CHK GRAY as shown in Fig. 9 may be run for the iterative method in order to retain the candidate objects which are relatively lighter than the background. To identify objects darker than the background, the image is inverted immediately after it has been generated.

[0163] The iterative method of the present invention also comprises the step of validating the candidate object having the valid object predetermined attribute value for each of the sub-histograms, thereby identifying the valid object. This step is shown in block F of Fig. 30. The validating step includes the sub-steps of calculating the candidate object attribute value and comparing the candidate object attribute value to the valid object predetermined attribute values to identify valid candidate objects. The validating step further includes the sub-step of checking for redundancies of the validate object to prevent multiple identification of the valid object. the redundancy checking sub-step is performed by the module CHK LIST as described above for the first embodiment with respect to Figs. 13A -D and 15A and 15B, and the module SET STAT as described above with respect to Figs. 14 and 16.

[0164] The iterative method of the third embodiment of the present invention further includes the step of performing a final check for redundancies of the valid object and resolving the redundancies to prevent multiple identification of the valid object. The final redundancy checking sub-step is performed by the module FINAL CHX as described above for the first embodiment with respect to Figs. 17A - B and the module INT STAT as illustrated in Fig. 18.

[0165] The iterative method of the third embodiment of the present invention further includes the step of filtering the image after the validation step. The filtering step for inhomogeneous objects is performed by the module CALCON as shown in Figs. 19A - D and for homogeneous objects is performed by the module CALCON as shown in Figs. 20A - B as described above with respect to the first embodiment.

[0166] The iterative method of the third embodiment of the present invention further includes the step of determining the number of valid objects in a candidate clump. The determining step is performed by the modules UNCLUMP and PROCESS CLUMP as shown in Figs. 21A - B and 22A - B, respectively, as described above with respect to the first embodiment. The iterative method of the third embodiment may be implemented in an image analysis system such as that shown in Fig. 24. Also, it will be apparent to those skilled in the art that various modifications may be made to the third embodiment of the present invention without departing from the scope and spirit of the invention. For instance, it is possible to use a co-occurrence matrix as described above in the second embodiment instead of the histogram in the iterative method to do adaptive thresholding.

[0167] In accordance with a fourth, embodiment of the present invention, also referred to as P-RETINA or parallel RETINA, there is provided a parallel processing method of identifying at least one valid object having at least one predetermined attribute defined by at least one predetermined attribute value in a background. The steps of the overall method of the fourth embodiment are illustrated in Fig. 32. The method according to the fourth embodiment of the present invention comprises the step of generating an image of the object and the background. An image is generated as shown in block A of Fig. 32. As in the first through third embodiments, the image of the object and the background may be generated by a camera. The image is then digitized and stored by a frame grabber or a video digitizer.

[0168] The method of the fourth embodiment of the present invention also comprises the step of generating a gray level histogram of the image. This step is shown in block B of Fig. 32. The module HISTOGRAM is run to perform this step. The steps in HISTOGRAM for generating the gray level histogram for the fourth embodiment are the same as those shown in the flow chart of Fig. 3.

[0169] The method of the fourth embodiment of the present invention also includes the step of selecting N global

entropic threshold gray levels as shown in block C of Fig. 32. The selecting step includes the sub-steps of sequentially partitioning the gray level histogram at each gray level into a first and a second partition. The entropy function is then computed for each partition. The entropy function of the histogram is defined as the sum of the entropy functions of the first and second partitions. An entropic threshold gray level is then selected such that the entropy function of the histogram is maximized. The gray level histogram is then subdivided using the entropically selected threshold gray level as defined above which maximizes the entropy function of the histogram as an upper delimiter and a lower delimiter to create an upper histogram and a lower histogram. The partitioning, computing and selecting steps are then repeated, where the repetition of the selecting step selects a next successive entropic threshold gray level. The subdividing step is then repeated using the next successive entropic threshold gray level as the entropic threshold gray level as defined in the selecting step to iteratively calculate the N global entropic threshold gray levels.

[0170] The method of the fourth embodiment of the present invention comprises the step of subdividing the gray level histogram into N + 1 sub-histograms using each of the global threshold gray levels as shown in block D. The N global entropic threshold gray levels and the N + 1 sub-histograms are the same as the those illustrated in Fig. 31 for the iterative method.

[0171] The method of the fourth embodiment of the present invention further includes the step of searching portions of the image corresponding to each sub-histogram using each of the N global entropically selected gray levels for at least one candidate object, where each candidate object has at least one candidate object attribute value. This step is shown in block E of Fig. 32. The searching step using the global entropically selected gray levels includes the sub-steps of scanning portions of the image corresponding to each sub-histogram using each global entropically selected threshold gray level for at least one candidate object and tracing the candidate object having boundary gray levels determined by each global entropically selected threshold gray level.

[0172] The method of the fourth embodiment of the present invention further includes the step of validating the candidate object found in the search using the global entropic threshold gray levels having the valid object predetermined attribute values to identify the valid object. This step is shown in block F of Fig. 32.

[0173] The parallel processing method of the present invention further includes the step of subdividing each sub-histogram into an upper sub-histogram and a lower sub-histogram using each global entropic threshold gray level as an upper delimiter and a lower delimiter. This step is shown in block G of Fig. 32.

[0174] The parallel processing method of the present invention further includes the step of selecting an entropic threshold gray level for each sub-histogram to maximize the entropy function of each of the upper and lower sub-histograms. This step is shown in block H of Fig. 32. The step of selecting an entropic threshold gray level for each sub-histogram to maximize the entropy function of each of the upper and lower sub-histograms includes the sub-steps of sequentially partitioning each gray level sub-histogram at each gray level into a first and a second partition and computing the entropy function for each partition. The entropy function of each sub-histogram is defined as the sum of the entropy functions of the first and second partitions. The selecting step further includes the sub-step of selecting an entropic threshold gray level such that the entropy function of each sub-histogram is maximized.

[0175] The method of the parallel processing embodiment of the present invention further includes the step of searching portions of the image corresponding to each sub-histogram using the entropically selected threshold gray level selected for each sub-histogram which maximizes the entropy function of each of the upper and lower sub-histograms for at least one candidate object. The step of searching portions of the image is shown in block I of Fig. 32. The searching step includes the sub-steps of scanning the portions of the image corresponding to each sub-histogram using each entropically selected threshold gray level for at least one candidate object and tracing the candidate object having boundary gray levels determined by each entropically selected threshold gray level.

[0176] The method according to the fourth embodiment of the present invention also comprises the step of validating the candidate object found in the search using the entropically selected threshold gray level selected for each sub-histogram which maximizes the entropy function thereof having the valid object predetermined attribute values, thereby identifying the valid object. This step is shown in block J of Fig. 32. Both the validating step for candidate objects found using the global entropic threshold gray levels and the entropic threshold gray levels include the sub-steps of calculating the candidate object attribute values and comparing the candidate object attribute values to the valid object predetermined attribute values to identify valid candidate objects. Both validating steps further include the sub-step of checking for redundancies of the valid object to prevent multiple identification of the valid object.

[0177] The steps of subdividing each sub-histogram, selecting an entropic threshold gray level for each sub-histogram, searching portions of the image corresponding to each sub-histogram using the entropically selected gray level of the selecting step for a candidate object and validating the candidate object are recursively repeated, where the repetition of the selecting step selects a next successive entropic threshold gray level, are then repeated. This step is shown in block K of Fig. 32. Each gray level sub-histogram is thereby recursively partitioned to identify valid objects until a predetermined minimum number of new valid objects is identified. As in the above-described embodiments, the predetermined minimum number may be zero.

[0178] The method according to the fourth embodiment of the present invention further includes the steps of per-

forming a final check for redundancies of the valid object and resolving the redundancies to prevent multiple identification of the valid object, filtering the image after the validation step and determining the number of valid objects in a candidate clump.

[0179] The parallel processing method of the fourth embodiment may be implemented in an image analysis system such as that shown in Fig. 24. Also, it will be apparent to those skilled in the art that various modifications can be made in the parallel processing method of the present invention without departing from the scope and spirit of the invention. For instance, it is possible to use a co-occurrence matrix as described above in the second embodiment instead of a histogram in the parallel processing method to do adaptive thresholding.

[0180] According to the fourth embodiment of the present invention, there is provided a system for identifying at least one valid object having at least one predetermined attribute value in a background. The system is shown generally at 30 in Fig. 33. The system includes means for generating an image of the object and the background. As shown in Fig. 33, the means for generating an image of the object and the background comprises a camera 32. Although a CCD camera is generally used with the present invention, any type of camera may be used without departing from the general principles of the present invention. A CCD camera, Model XC77, commercially available from Sony, Inc. of Cypress, California, has been proposed for use with the parallel processing system of the present invention.

[0181] The system according to the fourth embodiment also comprises means for digitizing and storing the image. The means for digitizing and storing the image comprises a frame grabber 34 as shown in Fig. 33. The frame grabber digitizes the video image and stores it in one frame, as known to one skilled in the image processing art. A frame grabber, Model 2851, which is commercially available from Data Translation, Inc. of Marlboro, Massachusetts, has been proposed for use with the fourth embodiment present invention.

[0182] Alternatively, the means for digitizing and storing the image may comprise a video digitizer, which also digitizes and stores the image, although not necessarily in one frame. The system of the present invention further comprises means for displaying the image. The means for displaying the image comprises a monitor 36 as shown in Fig. 33.

[0183] The system of the present invention also comprises computer means including a plurality of parallel processors. The plurality of parallel processors comprises a main processor 38a and at least one other parallel processor 38b as shown in Fig. 33. A 33MHz 386 PC computer, commercially available from Dell Computers of Austin, Texas, has been proposed for use with a Quadputer parallel processing board, commercially available from Microway of Kingston, Massachusetts, for the parallel processors of the present invention. Each parallel processor of the present invention comprises a driver 40, an entropic kernel 42 and a post-scan filter 46 as shown in Fig. 33. The driver stores the definition of a valid object. The entropic kernel of the main processor generates a gray level histogram of the image, selects N global entropic threshold gray levels, subdivides the gray level histogram into N + 1 sub-histograms, searches portions of the image corresponding to each sub-histogram using the global entropically selected gray levels, validates the candidate objects having the predetermined attribute values found in the search using the global entropically selected gray levels, and merges the valid objects having the predetermined attribute values found in the search using the global entropically selected gray levels and merges the valid objects found by all the parallel processors. The entropic kernel for the other processors subdivides each sub-histogram into upper and lower sub-histograms using each global entropic threshold gray level, selects an entropic threshold gray level to maximize the entropy function of each upper and lower sub-histogram, searches portions of the image corresponding to each sub-histogram using the entropically selected threshold gray level for at least one candidate object and validates the candidate object having the valid object predetermined attribute values found using the entropically selected threshold gray level. The entropic kernel for the other processors also recursively repeats the subdividing of the sub-histograms, the selection of an entropic threshold gray level, the searching using the entropically selected threshold gray level and the validating for the candidate objects found using the entropically selected threshold gray level to recursively partition each gray level sub-histogram until a predetermined minimum number of valid objects is identified.

[0184] Each parallel processor of the present invention may comprise software for performing the functions of the driver and the kernel as described above. Alternatively, each parallel processor may comprise a programmable read-only memory (PROM) from which the software may be retrieved. Each parallel processor of the configuration shown in Fig. 33 has a separate kernel and a separate driver. However, alternate configurations may be used in this embodiment without departing from the scope or spirit of the invention. For instance, each parallel processor may have a separate kernel, but all the processors may share a common driver. Alternatively, each processor may have a separate driver, but all the processors may share a common kernel. Finally, all the processors may share a common kernel and a common driver. In all of these configurations, each processor is dedicated to a specific window in gray level space and recursively searches for instances of valid objects within its appointed window.

[0185] The validated objects from each parallel processor are merged to one list as shown in box 48. The merging step involves performing one final redundancy check to prevent multiple identification of a valid object. The validated objects are represented by box 44 in Fig. 33. A post-scan filter is shown at 46 in Fig. 33 and provides a final check to remove redundancies in overlapping objects as described above.

[0186] According to a fifth embodiment of the present invention, there is provided a method of counting at least one

biological colony having at least one predetermined attribute value in a background. The steps of the overall method of the fifth embodiment are illustrated in Fig. 34.

[0187] The method of the fifth embodiment of the present invention comprises the step of generating an image of the colony and the background. This step is shown in block A of Fig. 34. As in the above embodiments, the image of the colony and the background may be generated by a camera. The image is then digitized and stored by a frame grabber or a video digitizer.

[0188] The method of the fifth embodiment of the present invention also comprises the step of generating a gray level histogram of the image. This step is shown in block B of Fig. 34. The module HISTOGRAM is run to perform this step. The steps in HISTOGRAM for generating the gray level histogram for the fifth embodiment are the same as those shown in the flow chart of Fig. 3.

[0189] The method of the fifth embodiment of the present invention also includes the step of entropically selecting a threshold gray level such that the entropy function of the histogram is maximized. This step is shown in block C of Fig. 34. The selecting step includes the sub-steps of sequentially partitioning the gray level histogram at each gray level into a first and a second partition. The entropy function is computed for each partition, where the entropy function of the histogram is defined as the sum of the entropy functions for the first and second partitions. A threshold gray level is then selected such that the entropy function of the histogram is maximized. The module ENTROPY is used to perform these sub-steps. The steps of the module ENTROPY are the same as those shown in Fig. 4.

[0190] The method of the fifth embodiment also comprises the step of searching the image for at least one candidate colony. This step is shown in block D of Fig. 34. The candidate colony has at least one candidate colony attribute value. The searching step includes the sub-steps of scanning the image for at least one candidate colony using the entropically selected threshold gray level and tracing the candidate colony having boundary gray levels determined by the entropically selected threshold gray level. The sub-steps of the searching step are performed by the module SEARCH IMAGE as shown in Fig. 6, FIND OBJECT as shown in Fig. 7 and TRACE OBJECT as shown in Fig. 8. The steps of these modules are the same as those shown in Figs. 6-8, except that the term "object" is replaced with the term "colony". The module CHK GRAY as shown in Fig. 9 with the term "object" replaced by the term "colony" may be run for the colony counting embodiment in order to retain the candidate colonies which are relatively lighter than the background. To identify colonies darker than the background, the image is inverted immediately after it has been generated.

[0191] The method the fifth embodiment of the present invention also comprises the step of validating the candidate colony having the valid colony predetermined attribute value to identify the valid colony. This step is shown by block E of Fig. 34 and is performed by the module TRACE OBJECT. The validating step includes the sub-steps of calculating the candidate colony attribute values and comparing the candidate colony attribute values to the valid colony predetermined attribute values to validate the candidate colony. The validating step further includes the sub-step of checking for redundancies of the validate colony to prevent multiple identification of the valid colony. The redundancy checking sub-step is performed by the module CHK LIST as described above for the first embodiment in Figs. 13A - D and 15A - B, and the module SET STAT as described with respect to Figs. 14 and 16, except that the term "object" in these flow charts is replaced with the term "colony".

[0192] The method of the fifth embodiment of the present invention also comprises the step of subdividing the gray level histogram into an upper histogram and a lower histogram using the entropically selecting threshold gray level which was previously selected to maximize the entropy function of the histogram as an upper delimiter and a lower delimiter. The method also comprises the step of recursively repeating the selecting, searching, validating and subdividing steps for each of the upper and lower histograms, where the repetition of the selecting step selects a next successive entropic threshold gray level. This repetition recursively partitions the gray level histogram to identify the valid colonies until a predetermined minimum number of new valid colonies is counted. This recursive repetition is performed by the module ANALYZE, which is run as shown in block F in Fig. 34. The steps of the ANALYZE module for colony counting are the same as those shown in Fig. 11, except that the term "object" in these flow charts is replaced with the term "colony".

[0193] The colony counting method of the fifth embodiment of the present invention further includes the step of performing a final check for redundancies of the valid colony and resolving the redundancies to prevent multiple identification of the valid colony. The final redundancy checking sub-step is performed by the module FINAL CHK as described above for the first embodiment with respect to Figs. 17A - B, and the module INT STAT as described with respect to Fig. 18, except that the term "object" in these flow charts is replaced with the term "colony".

[0194] The colony counting method of the fifth embodiment of the present invention further includes the step of filtering the image after the validation step. The filtering step for inhomogeneous colonies is performed by the module CALCON as shown in Figs. 19A - D and for homogeneous colonies is performed by the module CALCON as shown in Figs. 19A - B as described above with respect to the first embodiment, except that the term "object" in these flow charts is replaced with the term "colony".

[0195] The colony counting method according to the fifth embodiment of the present invention further includes the step of determining the number of valid colonies in a candidate clump. The determining step is performed by the module

UNCLUMP as shown in Figs. 21A - B and PROCESS CLUMP as shown in Figs. 22A - B and as described above with respect to the first embodiment, except that the term "object" in these flow charts is replaced with the term "colony".

[0196] The colony counting method of the present invention may be implemented in an image analysis system such as that shown in Fig. 24. Also, it will be apparent to those skilled in the art that various modifications can be made in the colony counting method of the present invention without departing from the scope and spirit of the invention. For instance, it is possible to use a co-occurrence matrix as described above in the second embodiment instead of a histogram in the colony counting method to do adaptive thresholding. In addition, the parallel processing method may be used to do colony counting.

[0197] According to a sixth embodiment of the present invention, there is provided a method of screening for at least one valid biological colony having at least one predetermined attribute value in a background. The steps of the screening method are illustrated in Fig. 35. The colony screening method employs the iterative techniques of the third embodiment as described above.

[0198] As shown in block A of Fig. 35, the colony screening method comprises the step of generating an image of the feature and the background. As in the above embodiments, the image of the colony and the background may be generated by a camera. The image is then digitized and stored by a frame grabber or a video digitizer.

[0199] The colony screening method of the present invention also comprises the step of generating a gray level histogram of the image. This step is shown in block B of Fig. 35. The module HISTOGRAM is run to perform this step. The steps in HISTOGRAM for generating the gray level histogram for the sixth embodiment are the same as those shown in the flow chart of Fig. 3.

[0200] The colony screening method of the present invention also includes the step of selecting N global entropic threshold gray levels. This step is shown in block C of Fig. 35. The selecting step includes the sub-steps of sequentially partitioning the gray level histogram at each gray level into a first and a second partition. The entropy function is computed for each partition, where the entropy function of the histogram is defined as the sum of the entropy functions for the first and second partitions. A global entropic threshold gray level is then selected such that the entropy function of the histogram is maximized. The selecting step further includes the sub-step of subdividing the gray level histogram using the global entropic threshold gray level which was used to maximize the entropy function of the histogram as an upper delimiter and a lower delimiter to create an upper histogram and a lower histogram. This step is shown in block C of Fig. 35. The partitioning, computing and selecting steps are then repeated for each of the upper and lower histograms, where the repetition of the selecting step selects a next successive global entropic threshold gray level. The subdividing step is then repeated using the next successive global entropic threshold gray level as the globally selected threshold gray level to iteratively calculate the N global entropic threshold gray levels. The module ENTROPY is used to perform these sub-steps. The steps of the module ENTROPY are the same as those shown in Fig. 4.

[0201] The colony screening method of the present invention also comprises the step of subdividing the original histogram into N + 1 histograms using the N global entropic threshold gray levels as shown in block D of Fig. 35.

[0202] The colony screening method of the present invention also comprises the step of searching portions of the image corresponding to each sub-histogram using each of the N global entropically selected threshold gray levels for at least one candidate colony. This step is shown in block E of Fig. 35. The candidate feature has at least one candidate colony attribute value. The searching step includes the sub-steps of scanning portions of the image corresponding to each sub-histogram for at least one candidate feature using the global entropically selected threshold gray level and tracing the candidate colony having a plurality of boundary gray levels determined by the entropically selected threshold gray level. The sub-steps of the searching step are performed by the module SEARCH IMAGE as shown in Fig. 6, FIND OBJECT as shown in Fig. 7 and TRACE OBJECT as shown in Fig. 8. The steps of these modules are the same as those shown in Figs. 6 - 8, except that the term "object" is replaced with the term "colony". The module CHK GRAY as shown in Fig. 9 with the term "object" replaced by the term "colony" may be run for the colony screening embodiment in order to retain the candidate colonies which are relatively lighter than the background. To identify colonies darker than the background, the image is inverted immediately after it has been generated.

[0203] The colony screening method of the present invention also comprises the step of validating the candidate colony having the valid colony predetermined attribute value to identify the valid colony. This step is shown by block F of Fig. 35. The validating step includes the sub-steps of calculating the candidate colony attribute values and comparing the candidate colony attribute values to the valid colony predetermined attribute values to validate the candidate colony. The validating step further includes the sub-step of checking for redundancies of the validate colony to prevent multiple identification of the valid colony. The redundancy checking sub-step is performed by the module CHK LIST as described above for the first embodiment in Figs. 13A - D and 15A - B, and the module SET STAT as described with respect to Figs. 14 and 16, except that the term "object" in these flow charts is replaced with the term "colony".

[0204] The colony screening method of the present invention further includes the step of performing a final check for redundancies of the valid colony and resolving the redundancies to prevent multiple identification of the valid colony. The final redundancy checking sub-step is performed by the module FINAL CHK as described above for the first embodiment in Figs. 17A - B and the module INT STAT as described with respect to Fig. 18, except that the term

"object" in these flow charts is replaced with the term "colony".

[0205] The colony screening method of the present invention further includes the step of filtering the image after the validation step. The filtering step for inhomogeneous objects is performed by the module CALCON as shown in Figs. 19A - D and for homogeneous objects is performed by the module CALCON as shown in Figs. 20A - B as described above with respect to the first embodiment, except that the term "object" in these flow charts is replaced with the term "colony".

[0206] The colony screening method according to the sixth embodiment of the present invention further includes the step of determining the number of valid colonies in a candidate clump. The determining step is performed by the module UNCLUMP as shown in Figs. 21A - B and the module PROCESS CLUMP as shown in Figs. 22A - B as described above with respect to the first embodiment, except that the term "object" in these flow charts is replaced with the term "colony".

[0207] The colony screening method of the present invention may be implemented in an image analysis system such as that shown in Fig. 24. Also, it will be apparent to those skilled in the art that various modifications can be made in the colony screening method of the present invention without departing from the scope and spirit of the invention. For instance, it is possible to use a co-occurrence matrix as described above in the second embodiment instead of a histogram in the colony screening method to adaptively segment an image of the colonies and the background. In addition, it is possible to use parallel processors to do colony screening.

[0208] According to a seventh embodiment of the present invention, there is provided a method of counting at least one discrete feature in a carpet, where the discrete feature has at least one predetermined attribute value. The steps of the overall method of the seventh embodiment are illustrated in Fig. 36.

[0209] As shown in block A of Fig. 36, the method comprises the step of generating an image of the feature and the carpet. As in the above embodiments, the image of the discrete feature and the carpet may be generated by a camera. The image is then digitized and stored by a frame grabber or a video digitizer.

[0210] The carpet discrete feature identification method of the seventh embodiment also comprises the step of generating a gray level histogram of the image. This step is shown in block B of Fig. 36. The module HISTOGRAM is run to perform this step. The steps in HISTOGRAM for generating the gray level histogram for the seventh embodiment are the same as those shown in the flow chart of Fig. 3.

[0211] The carpet feature identification method of the seventh embodiment also includes the step of entropically selecting a threshold gray level such that the entropy function of the histogram is maximized. This step is shown in block C of Fig. 36. The selecting step includes the sub-steps of sequentially partitioning the gray level histogram at each gray level into a first and a second partition. The entropy function is computed for each partition, where the entropy function of the histogram is defined as the sum of the entropy functions for the first and second partitions. A threshold gray level is then selected such that the entropy function of the histogram is maximized. The module ENTROPY is used to perform these sub-steps. The steps of the module ENTROPY are the same as those shown in Fig. 4.

[0212] The carpet feature identification method of the present invention also comprises the step of searching the image for at least one candidate feature. This step is shown in block D of Fig. 36. The candidate feature has at least one candidate feature attribute value. The searching step includes the sub-steps of scanning the image for at least one candidate feature using the entropically selected threshold gray level and tracing the candidate feature having boundary gray levels determined by the entropically selected threshold gray level. The sub-steps of the searching step are performed by the module SEARCH IMAGE as shown in Fig. 6, FIND OBJECT as shown in Fig. 7 and TRACE OBJECT as shown in Fig. 8. The steps of these modules are the same as those shown in Figs. 6 - 8, except that the term "object" is replaced with the term "feature". The module CHK GRAY as shown in Fig. 9 with the term "object" replaced by the term "feature" may be run for the carpet feature embodiment in order to retain the candidate features which are relatively lighter than the background. To identify features darker than the background, the image is inverted immediately after it has been generated.

[0213] The method of the embodiment of the present invention also comprises the step of validating the candidate feature having the valid feature predetermined attribute value to identify the valid feature. This step is shown by block E of Fig. 36 and is performed by the module TRACE OBJECT as shown in Fig. 8. The validating step includes the sub-steps of calculating the candidate feature attribute values and comparing the candidate feature attribute values to the valid feature predetermined attribute values to validate the candidate feature. The validating step further includes the sub-step of checking for redundancies of the validate feature to prevent multiple identification of the valid feature. The redundancy checking sub-step is performed by the module CHK LIST as described above for the first embodiment in Figs. 13A - D and 15A - B, and the module SET STAT as described with respect to Figs. 14 and 16, except that the term "object" is replaced with the term "feature".

[0214] The carpet feature identification method of the present invention also comprises the step of subdividing the gray level histogram into an upper histogram and a lower histogram using the entropically selecting threshold gray level which was previously selected to maximize the entropy function of the histogram as an upper delimiter and a lower delimiter. The method also comprises the step of recursively repeating the selecting, searching, validating and

subdividing steps for each of the upper and lower histograms, where the repetition of the selecting step selects a next successive entropic threshold gray level. This repetition recursively partitions the gray level histogram to identify the valid features until a predetermined minimum number of new valid features is counted. This recursive repetition is performed by the module ANALYZE, which is run as shown in block F in Fig. 36. The steps of the ANALYZE module for carpet feature identification are the same as those shown in Fig. 11, except that the term "object" in these flow charts is replaced with the term "feature".

[0215] The carpet feature identification method of the seventh embodiment of the present invention further includes the step of performing a final check for redundancies of the valid feature and resolving the redundancies to prevent multiple identification of the valid feature. The final redundancy checking sub-step is performed by the module FINAL CHK as described above for the first embodiment in Figs. 17A - B, and the module INT STAT as described with respect to Fig. 18, except that the term "object" in these flow charts is replaced with the term "feature".

[0216] The carpet feature identification method of the seventh embodiment of the present invention further includes the step of filtering the image after the validation step. The filtering step for inhomogeneous features is performed by the module CALCON as shown in Figs. 19A - D and for homogeneous features is performed by the module CALCON as shown in Figs. 20A - B as described above with respect to the first embodiment, except that the term "object" in these flow charts is replaced with the term "feature".

[0217] The carpet feature identification method of the present invention may be implemented in an image analysis system such as that shown in Fig. 24. Also, it will be apparent to those skilled in the art that various modifications can be made in the carpet feature identification method of the present invention without departing from the scope and spirit of the invention. For instance, it is possible to use a co-occurrence matrix as described above in the second embodiment instead of a histogram in the carpet feature identification method to adaptively segment an image of the features and the carpet. In addition, either the iterative or the parallel processing method may be used to do carpet feature identification.

[0218] According to an eighth embodiment of the present invention, there is provided a method of counting at least one pigment element embedded in a polymer. The pigment element has at least one predetermined attribute value. The steps of the overall method of the eighth embodiment are illustrated in Fig. 37.

[0219] As shown in block A of Fig. 37, the method comprises the step of generating an image of the pigment element and the polymer. As in the above embodiments, the image of the pigment element and the polymer may be generated by a camera. The image is then digitized and stored by a frame grabber or a video digitizer.

[0220] The method of the eighth embodiment of the present invention also comprises the step of generating a gray level histogram of the image. This step is shown in block B of Fig. 37. The module HISTOGRAM is run to perform this step. The steps in HISTOGRAM for generating the gray level histogram for the pigment element identification method are the same as those shown in the flow chart of Fig. 3.

[0221] The method of the eighth embodiment of the present invention also includes the step of entropically selecting a threshold gray level such that the entropy function of the histogram is maximized. This step is shown in block C of Fig. 37. The selecting step includes the sub-steps of sequentially partitioning the gray level histogram at each gray level into a first and a second partition. The entropy function is computed for each partition, where the entropy function of the histogram is defined as the sum of the entropy functions for the first and second partitions. A threshold gray level is then selected such that the entropy function of the histogram is maximized. The module ENTROPY is used to perform these sub-steps. The steps of the module ENTROPY are the same as those shown in Fig. 4.

[0222] The pigment element identification embodiment also comprises the step of searching the image for at least one candidate pigment element. This step is shown in block D of Fig. 37. The candidate pigment element has at least one candidate pigment element attribute value. The searching step includes the sub-steps of scanning the image for at least one candidate pigment element using the entropically selected threshold gray level and tracing the candidate pigment element having boundary gray levels determined by the entropically selected threshold gray level. The sub-steps of the searching step are performed by the module SEARCH IMAGE as shown in Fig. 6, FIND OBJECT as shown in Fig. 7 and TRACE OBJECT as shown in Fig. 8. The steps of these modules are the same as those shown in Figs. 6 - 8, except that the term "object" is replaced with the term "pigment element". The module CHK GRAY as shown in Fig. 9 with the term "object" replaced by the term "pigment element" may be run for the pigment element identification embodiment in order to retain the candidate pigment elements which are relatively lighter than the polymer. To identify pigments darker than the polymer, the image is inverted immediately after it has been generated.

[0223] The method of the pigment element identification embodiment of the present invention also comprises the step of validating the candidate element having the valid element predetermined attribute value to identify the valid pigment element. This step is shown by block E of Fig. 36 and is performed by the module TRACE OBJECT. The validating step includes the sub-steps of calculating the candidate pigment element attribute values and comparing the candidate pigment element attribute values to the valid pigment element predetermined attribute values to validate the candidate pigment element. The validating step further includes the sub-step of checking for redundancies of the valid pigment element to prevent multiple identification of the valid element. The redundancy checking sub-step is

performed by the module CHK LIST as described above for the first embodiment in Figs. 13A - D and 15A - B, and the module SET STAT as shown in Figs. 14 and 16, except that the term "object" in these flow charts is replaced with the term "pigment element".

[0224] The method of the pigment element identification embodiment of the present invention also comprises the step of subdividing the gray level histogram into an upper histogram and a lower histogram using the entropically selecting threshold gray level which was previously selected to maximize the entropy function of the histogram as an upper delimiter and a lower delimiter. The method also comprises the step of recursively repeating the selecting, searching, validating and subdividing steps for each of the upper and lower histograms, where the repetition of the selecting step selects a next successive entropic threshold gray level. This repetition recursively partitions the gray level histogram to identify the valid pigment elements until a predetermined minimum number of new valid pigment elements is counted. This repetition is performed by the module ANALYZE, which is run as shown in block F in Fig. 37. The steps of the ANALYZE module for pigment element identification are the same as those shown in Fig. 11, except that the term "object" is replaced with the term "pigment element".

[0225] The pigment element identification method of the present invention further includes the step of performing a final check for redundancies of the valid pigment element and resolving the redundancies to prevent multiple identification of the valid pigment element. The final redundancy checking sub-step is performed by the module FINAL CHK as described above for the first embodiment in Figs. 17A - B and the module INT STAT as shown in Fig. 18, except that the term "object" in these flow charts is replaced with the term "pigment element".

[0226] The pigment element identification method of the present invention further includes the step of filtering the image after the validation step. The filtering step for inhomogeneous pigment elements is performed by the module CALCON as shown in Figs. 19A - D and for homogeneous pigment elements is performed by the module CALCON as shown in Figs. 20A - B as described above with respect to the first embodiment, except that the term "object" in these flow charts is replaced with the term "pigment element".

[0227] The pigment element identification method of the present invention further may include the step of determining the number of valid pigment elements in a candidate clump. The determining step may be performed by the module UNCLUMP as shown in Figs. 21A - B and the module PROCESS CLUMP, as shown in Figs. 22A - B as described above with respect to the first embodiment, except that the term "object" in these flow charts is replaced with the term "pigment element".

[0228] The pigment element identification method of the present invention may be implemented in an image analysis system such as that shown in Fig. 24. Also, it will be apparent to those skilled in the art that various modifications can be made in the pigment element identification method of the present invention without departing from the scope and spirit of the invention. For instance, it is possible to use a co-occurrence matrix as described above in the second embodiment instead of a histogram in the pigment element identification method to do adaptive thresholding. In addition, either the iterative or the parallel processing method may be used to do pigment element identification.

[0229] The invention will be further clarified by the following examples, which are intended to be purely exemplary of the invention.

EXAMPLE I

[0230] In this Example, an application of the method and the image analysis system of the colony counting embodiment is described. Colony counting is a task routinely performed in microbiology laboratories for quantitating the amount of bacteria in a sample. The sample can be one of a number of different types such as blood, food, cosmetics, soil, etc. Colony counting has been performed using the present system to identify and quantitate:

- (i) aerobic bacteria on a BHI (Brain Heart Infusion) agar plate (inhomogeneous colonies), and
- (ii) Salmonella colonies on a Hektoen Enteric agar plate (homogeneous colonies).

[0231] In addition, the system is capable of identifying and quantitating different species of bacteria on a plate containing more than one species. This is useful for identifying pathogenic bacteria in plate containing both pathogenic and non-pathogenic organisms.

[0232] The apparatus used in this Example to count colonies comprised a Petri dish holder to hold the plate, an optical system for illuminating the plate and an image acquisition system to generate an image of the sample on the plate as described above with respect to Fig. 24. A video signal from a camera was fed into a frame grabber which occupies a slot in a PC. The frame grabber digitized the image.

[0233] The Petri dish holder comprised a movable, machined member which can accept standard Petri dishes. The member had a countersunk circular indent capable of accepting the dishes and was slid into place via a bearing-based support. Below the holder, there was room to insert an appropriate background sheet to enhance image contrast.

[0234] The illumination source used comprised a Philips FC 8T9/WW circular fluorescent lamp (8" diameter) which

surrounded the Petri dish. The lamp was mounted approximately 1" above the dish and illuminated the dish uniformly from all sides. The distance between the lamp and the dish is important. If the lamp is too high, the resulting image will reveal significant "glaring" and "haloing" from the curved surfaces of the colonies. The colonies can be considered small 3D lenses, and oblique, grazing angle illumination is necessary so that light scattering from the lens-like surfaces does not enter the camera. The lamp was actively stabilized using a Mercron FL0416-2 controller which kept the light level constant, to prevent fluctuations in output. In addition, the controller was remotely programmable to adjust the brightness of the lamp. This feature is especially useful for colony plates where the inherent contrast is variable. For low contrast plates, the light level can be increased to improve the contrast.

[0235] The optical system comprised a Sony XC-77 monochrome CCD video camera with a 16mm F2.8 Nikkor lens. The camera was centered above the Petri dish at a distance of approximately 1.5 feet. The distance from the camera to the dish was such that the disk just filled the surface of the CCD sensor in order to maximize spatial resolution. A black DELRIN light shield extended from the camera to the plate to prevent stray light from entering the camera. The video signal from the camera went directly to the frame grabber on the PC.

[0236] The image acquisition system comprised a Data Translation DT2851 frame grabber board which occupied an expansion slot on the PC. A Dell 33 MHZ 386AT system was used to store images and execute the colony screening method of the present invention. The frame grabber board had a 512 x 512 pixel video buffer in which the digitized image was stored. In order to minimize "shot" noise from the camera, several image frames were averaged, and the resulting averaged image was stored and used for further analysis. The output of the frame grabber was connected to a Sony TRINITRON monitor to display the image.

[0237] The attributes for valid colonies as defined in the driver for this application are shown in Table 1. The range of attribute values for inhomogeneous colonies is shown in Table 2. The range of attribute values for homogeneous colonies is shown in Table 3.

[0238] The three-prong filter described above for filtering inhomogeneous colonies was used for sample (i) above. The artifact removal filter described for homogeneous objects in the specification was used for filtering sample (ii) above. The redundancy check for inhomogeneous objects described above was used for redundancy checking in sample (i). The redundancy check for homogeneous objects described above was used for redundancy checking in sample (ii).

[0239] For the BHI plates, 150 plates were processed with the image analysis system of the fifth embodiment, where the colony population ranged from 0-350 and compared to a manual counting of the same plates. The square of the linear correlation coefficient for a least squares linear fit for a correlation curve for this data was shown to be 0.981. The average difference between the method according to the fifth embodiment of the present invention and manual counting was approximately 8%.

Table 1

| Attributes of objects | |
|-----------------------|---|
| (1) | xcm = x coordinate of object center of mass |
| (2) | ycm = y coordinate of object center of mass |
| (3) | perim = total length of perimeter of object |
| (4) | npts = number of perimeter points in image |
| (5) | obj_type = type of object (Exterior or Interior) Exterior = object never contained within another object Interior = object previously contained within another object |
| (6) | area = area of object |
| (7) | shape = shape factor of object |
| (8) | status = status value of object which is used in artifact removal and redundancy checking in several different ways |
| (9) | contrast = edge contrast of object |
| (10) | edge = data structure containing coordinates of extremum points of object |
| (11) | thresh = threshold at which object was detected |
| (12) | max = maximum gray level of histogram partition in which object was detected |
| (13) | clump_shape = maximum shape factor for a candidate clumped object |
| (14) | clump_size = minimum area for a clump of candidate objects |
| (15) | worst_shape = minimum shape factor for a clump of candidate objects |

Table 2

| Range of Attribute values for inhomogeneous colonies | | |
|---|---|--------------------|
| Note: Only some of the attribute values require valid ranges. Others are "descriptors" which do not influence object validity. | | |
| Attribute | | Valid Range |
| (1) | xcm = x coordinate of object center of mass | Descriptor |
| (2) | ycm = y coordinate of object center of mass | Descriptor |
| (3) | perim = total length of perimeter of object | Descriptor |
| (4) | npts = number of perimeter points in image | npts > 5 |
| (5) | obj type = type of object (Exterior or Interior) Exterior = object never contained within another object Interior = object previously contained within another object | Descriptor |
| (6) | area = area of object | area > 0.0 |
| (7) | shape = shape factor of object | shape > -0.8 |
| (8) | status = status value of object which is used in artifact removal and redundancy checking in several different ways | DESCRIPTOR |
| (9) | contrast = edge contrast of object | contrast > 125 |
| (10) | edge = data structure containing coordinates of extremum points of object | DESCRIPTOR |
| (11) | thresh = threshold at which object was detected | DESCRIPTOR |
| (12) | max = maximum gray level of histogram partition in which object was detected | DESCRIPTOR |
| (13) | clump shape = maximum shape factor for a clump of candidate objects | clump shape = -0.8 |
| (14) | clump size = minimum area for a clump of candidate objects | clump size = 50.0 |
| (15) | worst shape = minimum shape factor for a clump of candidate objects | worst shape = -5.0 |

Table 3

| Range of Attribute values for homogeneous colonies | | |
|---|---|--------------------|
| Note: Only some of the attribute values require valid ranges. Others are "descriptors" which do not influence object validity. | | |
| Attribute | | Valid Range |
| (1) | xcm = x coordinate of object center of mass | Descriptor |
| (2) | ycm = y coordinate of object center of mass | Descriptor |
| (3) | perim = total length of perimeter of object | Descriptor |
| (4) | npts = number of perimeter points in object | npts > 3 |
| (5) | obj type = type of object (Exterior or Interior) Exterior = object never contained within another object Interior = object previously contained within another object | Descriptor |
| (6) | area = area of object | area > 0.0 |
| (7) | shape = shape factor of object | shape > -0.8 |
| (8) | status = status value of object which is used in artifact removal and redundancy checking in several different ways | DESCRIPTOR |
| (9) | contrast = edge contrast of object | contrast > 125 |
| (10) | edge = data structure containing coordinates of extremum points of object | DESCRIPTOR |
| (11) | thresh = threshold at which object was detected | DESCRIPTOR |
| (12) | max = maximum gray level of histogram partition in which object was detected | DESCRIPTOR |
| (13) | clump shape = maximum shape factor for a clump of candidate objects | clump shape = -0.3 |
| (14) | clump size = minimum area for a clump of candidate objects | clump size = 30.0 |
| (15) | worst shape = minimum shape factor for a clump of candidate objects | worst shape = -5.0 |

EXAMPLE II

[0240] In this Example, an application of the method and image analysis system of the colony screening embodiment is described. Colony screening is a task routinely performed in microbiology laboratories for identifying a specific type of bacteria in a sample. The sample can be one of a number of different types such as blood, food, cosmetics, soil, etc. The screening process involves "streaking" an agar plate with the sample of interest. As the sample is streaked across the plate, it effectively gets diluted, resulting in a wide range of sample concentrations over the plate. The question to be answered is whether or not the sample contains a pathogenic organism. The screening process results in a yes-no answer; absolute quantitation is not required. For this Example, the sample of interest was food, where a screening step was used to identify potentially contaminated food samples. By flagging contaminated samples, it was possible to perform effective quality control as well as separating out the contaminated food for more detailed analysis. The apparatus used to screen colonies was set up as described in Example I.

[0241] To illustrate the colony screening system, an example of a "streak" plate with Salmonella colonies on a Hektoen Enteric agar plate was provided. The goal was to flag this plate as a contaminated plate; absolute quantitation was not required. The complexity of this plate is shown in the varying background introduced in the streaking process. The attributes for valid colonies as defined in the driver for this Example are the same as those shown in Table 1. The range of attribute values for screening is shown in Table 4.

Table 4

| Range of Attribute values for screening | | |
|---|---|----------------|
| Note: Only some of the attribute values require valid ranges. Others are "descriptors" which do not influence object validity. | | |
| Attribute | | Valid Range |
| (1) | xcm = x coordinate of object center of mass | Descriptor |
| (2) | ycm = y coordinate of object center of mass | Descriptor |
| (3) | perim = total length of perimeter of object | Descriptor |
| (4) | npts = number of perimeter points in object | npts > 3 |
| (5) | obj type = type of object (Exterior or Interior) | Descriptor |
| | Exterior = object never contained within another object | |
| | Interior = object previously contained within another object | |
| (6) | area = area of object | area > 0.0 |
| (7) | shape = shape factor of object | shape > -0.8 |
| (8) | status = status value of object which is used in artifact removal and redundancy checking in several different ways | DESCRIPTOR |
| (9) | contrast = edge contrast of object | contrast > 100 |
| (10) | edge = data structure containing coordinates of extremum points of object | DESCRIPTOR |
| (11) | thresh = threshold at which object was detected | DESCRIPTOR |
| (12) | max = maximum gray level of histogram partition in which object was detected | DESCRIPTOR |
| (13) | clump shape = maximum shape factor for a clump of candidate objects | UNUSED |
| (14) | clump size = minimum area for a clump of candidate objects | UNUSED |
| (15) | worst shape = minimum shape factor for a clump of candidate objects | UNUSED |

[0242] For the screening Example, the artifact removal filter algorithm comprised two steps:

- (i) a candidate colony was deleted if it had an area less than a predetermined area; and
- / (ii) a candidate colony was deleted if it had an edge contrast less than a predetermined minimum edge contrast.

[0243] The redundancy checking sub-steps for homogeneous objects described above was used for redundancy checking in this Example, with the proviso that for overlapping objects, the criterion used for determining which object to retain was always to keep the larger object. There was no unclumping performed in the screening method of this Example.

[0244] On a sample set of 99 "streak" plates, 89 plates analyzed by IETNA agreed with manual determination of whether the plates were positive (i.e., contained the Salmonella organism) or negative (i.e., did not contain the Salmonella organism). There were four "false negatives" where IETNA as used for colony screening failed to flag any Salmonella organisms on plates where it was manually determined that such organisms were present. There were six

"false positives" in this Example, where IETNA as used for colony screening flagged plates as containing Salmonella organisms. This is contrasted with manual determination which indicated that there were no such organisms present.

EXAMPLE III

[0245] In this Example, an application of the image analysis system and method of the carpet feature identification embodiment is described. A major factor influencing the appearance of carpets is the presence of discrete features, such as defects, on the carpet surface. Defects such as streaks, "chevrons" (which have a chevron like shape and appear fairly periodically across the carpet surface) can result from problems in the carpet processing cycle and need to be identified in a timely fashion to maintain high-quality product.

[0246] The present invention has been adapted towards this end as a general purpose "Discrete Feature Analyzer", where a variety of defects can be identified on the complex background of a carpet surface. For a specific defect, the attributes of the defect can be defined in the driver, and the present invention will then identify all instances of that particular defect.

[0247] The carpet sample used for analysis was a Suessen set, fluid-dyed, two-ply, 1150 denier sample obtained from DuPont flooring systems. This sample had several "blob" like defects distributed over the surface which needed to be identified.

[0248] The apparatus used to generate the carpet images consisted of a Macbeth Spectra Light SPL-65 light box, a Sony XC-77 CCD camera, a Data Translation 2851 frame grabber board and a PC. The light box ensured that the spectral energy distribution of the illuminating light matched the spectral energy distribution of sunlight. The camera viewed the carpet sample at an oblique angle to maximize the contrast of the defects.

[0249] The attributes for valid defects as defined in the driver are the same as those shown in Table 1. The range of attribute values for valid defects is shown in Table 5.

Table 5

| Range of Attribute values for carpets | | |
|---|--|---------------|
| <u>Note:</u> Only some of the attribute values require valid ranges. Others are "descriptors" which do not influence object validity. | | |
| Attribute | | Valid Range |
| (1) xcm = x coordinate of object center of mass | | Descriptor |
| (2) ycm = y coordinate of object center of mass | | Descriptor |
| (3) perim = total length of perimeter of object | | Descriptor |
| (4) npts = number of perimeter points in object | | npts > 10 |
| (5) obj type = type of object (Exterior or Interior) | | Descriptor |
| Exterior = object never contained within another object | | |
| Interior = object previously contained within another object | | |
| (6) area = area of object | | area > 0.0 |
| (7) shape = shape factor of object | | shape > -10.0 |
| (8) status = status value of object which is used in artifact removal and redundancy checking in several different ways | | DESCRIPTOR |
| (9) contrast = edge contrast of object | | contrast > 0 |
| (10) edge = data structure containing coordinates of extremum points of object | | DESCRIPTOR |
| (11) thresh = threshold at which object was detected | | DESCRIPTOR |
| (12) max = maximum gray level of histogram partition in which object was detected | | DESCRIPTOR |
| (13) clump shape = maximum shape factor for a clump of candidate objects | | UNUSED |
| (14) clump size = minimum area for a clump of candidate objects | | UNUSED |
| (15) worst shape = minimum shape factor for a clump of candidate objects | | UNUSED |

[0250] For identifying defects, the artifact removal comprised three steps:

- (i) a candidate defect was deleted if it had an area less than a predetermined area;
- (ii) a candidate defect was deleted if it had an edge contrast less than a predetermined minimum edge contrast;
- (iii) a candidate defect was deleted if it had a shape factor less than a predetermined shape factor.

[0251] The redundancy check for homogeneous defects described above was used for redundancy checking in this

Example with a proviso that for overlapping defects, the criterion used for determining which defects to retain was always to keep the larger defects. There was no unclumping performed in this Example.

EXAMPLE IV

[0252] In this Example, an application of the method and system of the pigment element identification embodiment of the present invention is described. An important element in producing carpet fibers with uniform appearance is the amount of mixing of colored pigment elements within the fiber. To perform a quantitative study of the mixing of pigment elements with a molten polymer, it is important to be able to calculate the spatial and size distributions of the pigment elements across cross-sections of polymer emerging from a static mixer. The system of the eighth embodiment of the present invention is capable of performing these calculations by automatically identifying the pigment elements across the cross-section of the polymer.

[0253] The sample of interest contained a distribution of iron oxide pigment elements in a NRD159 carrier. NRD159 is a low molecular weight nylon derivative commercially available from E. I. du Pont de Nemours and Company of Wilmington, Delaware. The image of the pigment distribution was obtained from a micro-toned piece of polymer imaged through a Nikon microscope with a 20x objective. The image was focused onto a Sony XC-77 CCD camera which sent a video signal to a DEC MicroVax computer. The image was then digitized using a Data Translation DT2651 Frame Grabber board.

[0254] The attributes for valid pigments as defined in the driver are the same as shown in Table 1. The range of attribute values for valid objects (pigments) is shown in Table 6.

Table 6

| Range of Attribute values for pigments | | |
|---|--|--------------|
| Note: Only some of the attribute values require valid ranges. Others are "descriptors" which do not influence object validity. | | |
| Attribute | | Valid Range |
| (1) xcm = x coordinate of object center of mass | | Descriptor |
| (2) ycm = y coordinate of object center of mass | | Descriptor |
| (3) perim = total length of perimeter of object | | Descriptor |
| (4) npts = number of perimeter points in object | | npts > 3 |
| (5) obj type = type of object (Exterior or Interior) | | Descriptor |
| Exterior = object never contained within another object | | |
| Interior = object previously contained within another object | | |
| (6) area = area of object | | area > 0.0 |
| (7) shape = shape factor of object | | shape > -7.0 |
| (8) status = status value of object which is used in artifact removal and redundancy checking in several different ways | | DESCRIPTOR |
| (9) contrast = edge contrast of object | | contrast > 0 |
| (10) edge = data structure containing coordinates of extremum points of object | | DESCRIPTOR |
| (11) thresh = threshold at which object was detected | | DESCRIPTOR |
| (12) max = maximum gray level of histogram partition in which object was detected | | DESCRIPTOR |
| (13) clump shape = maximum shape factor for a clump of candidate objects | | UNUSED |
| (14) clump size = minimum area for a clump of candidate objects | | UNUSED |
| (15) worst shape = minimum shape factor for a clump of candidate objects | | UNUSED |

[0255] For pigment identification, the artifact removal filter algorithm used was the same as that used for the screening process of Example II. The redundancy check algorithm for pigment element identification was the same as that used for the screening process of Example II.

[0256] The adaptive nature of the invention allowed it to identify all the pigment elements. However, it would be possible to identify a subset of the pigment elements with a certain contrast range if so desired. This would effectively allow the modules of the present invention to "section" through various focal planes of material to identify pigment elements at different depths.

[0257] Other embodiments of the invention will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only.

Claims

1. A method of identifying at least one valid object having at least one predetermined attribute value in a background, comprising the steps of:
 - (a) generating an image of the object and the background;
 - (b) defining a data space representative of the image, the data space comprising the range of all values of all parameters of the image;
 - (c) generating a list of ordered gray level or color threshold pairs;
 - (d) subdividing the data space into a plurality of sub-spaces by using the threshold pairs to define each sub-space as a sub-range of values of the parameters of the image;
 - (e) selecting at least one sub-space in which to search for a candidate object;
 - (f) searching the image multiple times using each selected sub-space, wherein a candidate object has at least one attribute value; and
 - (g) validating the candidate object to identify the valid object by comparing the at least one attribute value of each candidate object to the predetermined at least one attribute value of the valid object.
2. The method of claim 1, wherein the step of searching the image comprises scanning the image once using each of the sub-spaces simultaneously.
3. The method of claim 1, wherein the step of searching the image comprises scanning the image multiple times using a selected sub-space for each scan.
4. The method of claim 1, wherein the data space comprises a color space of the image.
5. The method of claim 1, wherein the data space comprises a space resulting from the transformation of pixel values of an image.
6. The method of Claim 1, wherein the step of generating the thresholds comprises generating a gray level histogram of the image, where the gray level histogram has an entropy function.
7. The method of Claim 6, wherein the step of generating thresholds further comprises entropically selecting a threshold gray level such that the entropy function of the histogram is maximized.
8. The method of claim 7, wherein the step of entropically selecting a threshold pixel value includes the sub-steps of:
 - (i) sequentially partitioning the pixel value histogram at each pixel value into a first partition and a second partition, and
 - (ii) computing the entropy function for each partition, where the total entropy function of the histogram is defined as the sum of the entropy function of the first partition and the entropy function of the second partition.
9. A method of identifying at least one valid object having at least one predetermined attribute value in a background, comprising the steps of:
 - (a) generating an image of the object and the background;
 - (b) defining a data space representative of the image, the data space comprising the range of all values of the parameters of the image;
 - (c) generating a list of ordered gray level or color threshold pairs;
 - (d) subdividing the data space into an upper sub-space and a lower sub-space;
 - (e) recursively repeating the step of subdividing the data space into an upper sub-space and a lower sub-space;
 - (f) selecting at least one upper or lower sub-space;
 - (g) searching the image multiple times using each selected sub-space, for at least one representation of a candidate object, wherein the candidate object has at least one attribute value; and
 - (h) validating the candidate object to identify the valid object by comparing the at least one attribute value of each candidate object to the predetermined at least one attribute value of the valid object, wherein the repetition of the selection step selects the next successive sub-space, thereby recursively partitioning the data space until a condition for terminating the multiple searching has been reached.

10. The method of claim 9, wherein the terminating condition is that a predetermined number of new valid objects is identified.
11. The method of claim 10, wherein the predetermined number of new valid objects is greater than zero.
12. The method of claim 10, wherein the predetermined number of new valid objects is zero.
13. The method of claim 9, wherein the terminating condition is that a minimum pixel level partition width has been reached during the recursive partitioning of the data space.

Patentansprüche

1. Verfahren zur Identifikation mindestens eines gültigen Objekts mit mindestens einem vorgegebenen Attributwert vor einem Hintergrund, mit den folgenden Schritten:
- (a) Erzeugen eines Bildes des Objekts und des Hintergrunds;
 - (b) Definieren eines das Bild darstellenden Datenraums, wobei der Datenraum den Bereich aller Werte aller Parameter des Bildes aufweist;
 - (c) Erzeugen einer Liste von geordneten Graustufen- oder Farbschwellwertpaaren;
 - (d) Unterteilen des Datenraums in mehrere Unterräume unter Verwendung der Schwellwertpaare zur Definition jedes Unterrums als Unterbereich von Parameterwerten des Bildes;
 - (e) Auswahl mindestens eines Unterrums, in dem nach einem Anwarter gesucht werden soll;
 - (f) mehrmaliges Durchsuchen des Bildes unter Verwendung jedes ausgewählten Unterrums, wobei ein Anwarterobjekt mindestens einen Attributwert aufweist; und
 - (g) Bestätigen des Anwarterobjekts zu Identifikation des gültigen Objekts durch Vergleich mindestens eines Attributwerts jedes Anwarterobjekts mit dem vorgegebenen mindestens einen Attributwert des gültigen Objekts.
2. Verfahren nach Anspruch 1, wobei der Schritt zum Durchsuchen des Bildes das einmalige Abtasten des Bildes unter gleichzeitiger Verwendung jedes der Unterräume aufweist.
3. Verfahren nach Anspruch 1, wobei der Schritt zum Durchsuchen des Bildes das mehrmalige Abtasten des Bildes unter Verwendung eines ausgewählten Unterrums für jede Abtastung aufweist.
4. Verfahren nach Anspruch 1, wobei der Datenraum einen Farbraum des Bildes aufweist.
5. Verfahren nach Anspruch 1, wobei der Datenraum einen Raum aufweist, der aus der Transformation der Pixelwerte eines Bildes resultiert.
6. Verfahren nach Anspruch 1, wobei der Schritt der Schwellwerterzeugung die Erzeugung eines Graustufenhistogramms des Bildes aufweist, wobei das Graustufenhistogramm eine Entropiefunktion aufweist.
7. Verfahren nach Anspruch 6, wobei der Schritt der Schwellwerterzeugung ferner die entropische Auswahl einer Schwellwert-Graustufe aufweist, derart daß die Entropiefunktion des Histogramms maximiert wird.
8. Verfahren nach Anspruch 7, wobei der Schritt der entropischen Auswahl eines Pixelschwellwerts die folgenden Teilschritte aufweist:
- (i) sequentielles Unterteilen des Pixelwert-Histogramms in jedem Pixelwert in einen ersten Bereich und einen zweiten Bereich, und
 - (ii) Berechnen der Entropiefunktion für jeden Bereich, wobei die Gesamt-Entropiefunktion des Histogramms als Summe der Entropiefunktion des ersten Bereichs und der Entropiefunktion des zweiten Bereichs definiert ist.
9. Verfahren zur Identifikation mindestens eines gültigen Objekts mit mindestens einem vorgegebenen Attributwert vor einem Hintergrund, mit den folgenden Schritten:

- (a) Erzeugen eines Bildes des Objekts und des Hintergrunds;
 - (b) Definieren eines das Bild darstellenden Datenraums, wobei der Datenraum den Bereich aller Werte der Parameter des Bildes aufweist;
 - (c) Erzeugen einer Liste von geordneten Graustufen- oder Farbschwellwertpaaren;
 - (d) Unterteilen des Datenraums in einen oberen Unterraum und einen unteren Unterraum;
 - (e) rekursives Wiederholen des Schritts der Unterteilung des Datenraums in einen oberen Unterraum und einen unteren Unterraum;
 - (f) Auswahl mindestens eines oberen oder unteren Unterraums;
 - (g) mehrmaliges Durchsuchen des Bildes unter Verwendung jedes ausgewählten Unterraums nach mindestens einer Darstellung eines Anwörterobjekts, wobei das Anwörterobjekt mindestens einen Attributwert aufweist; und
 - (h) Bestätigen des Anwörterobjekts zu Identifikation des gültigen Objekts durch Vergleich des mindestens einen Attributwerts jedes Anwörterobjekts mit dem vorgegebenen mindestens einen Attributwert des gültigen Objekts, wobei die Wiederholung des Auswahl schritts den nächstfolgenden Unterraum auswählt, wodurch der Datenraum rekursiv unterteilt wird, bis eine Abbruchbedingung der Mehrfachsuche erreicht worden ist.
10. Verfahren nach Anspruch 9, wobei die Abbruchbedingung ist, daß eine vorgegebene Anzahl neuer gültiger Objekte identifiziert wird.
11. Verfahren nach Anspruch 10, wobei die vorgegebene Anzahl neuer gültiger Objekte größer als null ist.
12. Verfahren nach Anspruch 10, wobei die vorgegebene Anzahl neuer gültiger Objekte gleich null ist.
13. Verfahren nach Anspruch 9, wobei die Abbruchbedingung ist, daß während der rekursiven Unterteilung des Datenraums eine minimale Pixelebenen-Unterteilungsbreite erreicht worden ist.

Revendications

1. Procédé d'identification d'au moins un objet valide présentant au moins une valeur d'attribut prédéterminée dans un fond, comprenant les étapes de:
- (a) génération d'une image de l'objet et du fond;
 - (b) définition d'un espace de données représentatif de l'image, l'espace de données comprenant la plage de toutes les valeurs de tous les paramètres de l'image;
 - (c) génération d'une liste de paires de seuils de niveau de gris ou de couleur ordonnés;
 - (d) subdivision de l'espace de données selon une pluralité de sous-espaces en utilisant les paires de seuils afin de définir chaque sous-espace en tant que sous-plage de valeurs des paramètres de l'image;
 - (e) sélection d'au moins un sous-espace dans lequel il convient de rechercher un objet candidat;
 - (f) recherche dans l'image de multiples fois en utilisant chaque sous-espace sélectionné où un objet candidat présente au moins une valeur d'attribut; et
 - (g) validation de l'objet candidat afin d'identifier l'objet valide en comparant l'au moins une valeur d'attribut de chaque objet candidat à l'au moins une valeur d'attribut prédéterminée de l'objet valide.
2. Procédé selon la revendication 1, dans lequel l'étape de recherche dans l'image comprend le balayage de l'image une fois en utilisant chacun des sous-espaces simultanément.
3. Procédé selon la revendication 1, dans lequel l'étape de recherche dans l'image comprend le balayage de l'image de multiples fois en utilisant un sous-espace sélectionné pour chaque balayage.
4. Procédé selon la revendication 1, dans lequel l'espace de données comprend un espace couleur de l'image.
5. Procédé selon la revendication 1, dans lequel l'espace de données comprend un espace qui résulte de la transformation de valeurs de pixel d'une image.
6. Procédé selon la revendication 1, dans lequel l'étape de génération de seuils comprend la génération d'un histogramme de niveaux de gris de l'image où l'histogramme de niveaux de gris présente une fonction d'entropie.

7. Procédé selon la revendication 6, dans lequel l'étape de génération de seuils comprend en outre la sélection de façon entropique d'un niveau de gris de seuil de telle sorte que la fonction d'entropie de l'histogramme soit maximisée.
- 5 8. Procédé selon la revendication 7, dans lequel l'étape de sélection de façon entropique d'une valeur de pixel de seuil inclut les sous-étapes de:
- (i) partitionnement de façon séquentielle de l'histogramme de valeurs de pixel au niveau de chaque valeur de pixel selon une première partition et une seconde partition; et
- 10 (ii) calcul de la fonction d'entropie pour chaque partition où la fonction d'entropie totale de l'histogramme est définie en tant que somme de la fonction d'entropie de la première partition et de la fonction d'entropie de la seconde partition.
9. Procédé d'identification d'au moins un objet valide présentant au moins une valeur d'attribut prédéterminée dans un fond, comprenant les étapes de:
- 15 (a) génération d'une image de l'objet et du fond;
- (b) définition d'un espace de données représentatif de l'image, l'espace de données comprenant la plage de toutes les valeurs des paramètres de l'image;
- 20 (c) génération d'une liste de paires de seuils de niveau de gris ou de couleur ordonnés;
- (d) subdivision de l'espace de données selon un sous-espace supérieur et un sous-espace inférieur;
- (e) répétition de façon récursive de l'étape de subdivision de l'espace de données selon un sous-espace supérieur et un sous-espace inférieur;
- (f) sélection d'au moins un sous-espace pris parmi le sous-espace supérieur et le sous-espace inférieur;
- 25 (g) recherche, dans l'image de multiples fois en utilisant chaque sous-espace sélectionné, d'au moins une représentation d'un objet candidat où l'objet candidat présente au moins une valeur d'attribut; et
- (h) validation de l'objet candidat afin d'identifier l'objet valide en comparant l'au moins une valeur d'attribut de chaque objet candidat à l'au moins une valeur d'attribut prédéterminée de l'objet valide où la répétition de l'étape de sélection sélectionne le sous-espace successif suivant, d'où ainsi le partitionnement de façon ré-
- 30 cursive de l'espace de données jusqu'à ce qu'une condition pour achever la recherche multiple ait été atteinte.
10. Procédé selon la revendication 9, dans lequel la condition d'achèvement est constituée par le fait qu'un nombre prédéterminé de nouveaux objets valides est identifié.
- 35 11. Procédé selon la revendication 10, dans lequel le nombre prédéterminé de nouveaux objets valides est supérieur à zéro.
12. Procédé selon la revendication 10, dans lequel le nombre prédéterminé de nouveaux objets valides est de zéro.
- 40 13. Procédé selon la revendication 9, dans lequel la condition d'achèvement est constituée par le fait qu'une largeur de partition de niveau de pixel minimum a été atteinte pendant le partitionnement récursif de l'espace de données.

45

50

55

FIG. 1

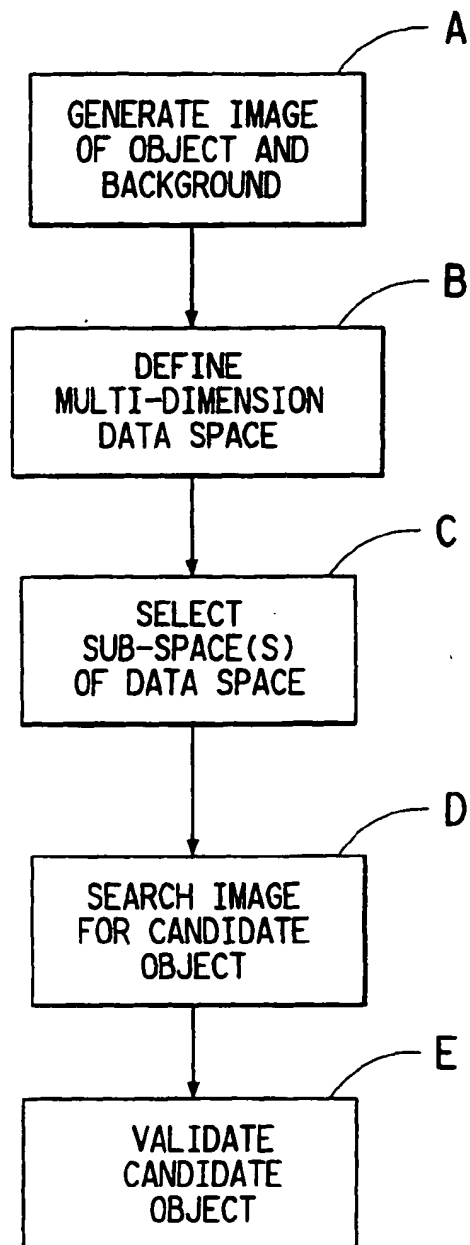


FIG.2

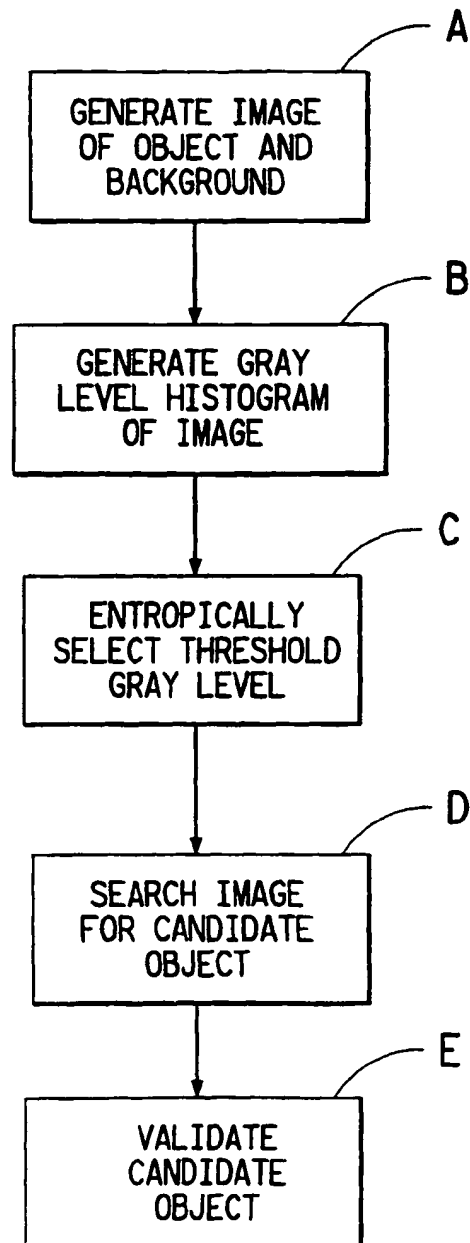


FIG. 3

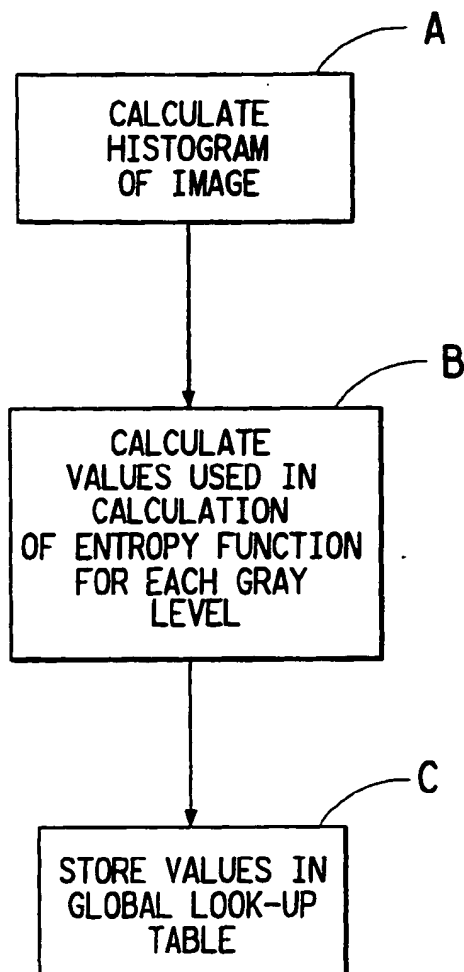


FIG. 4

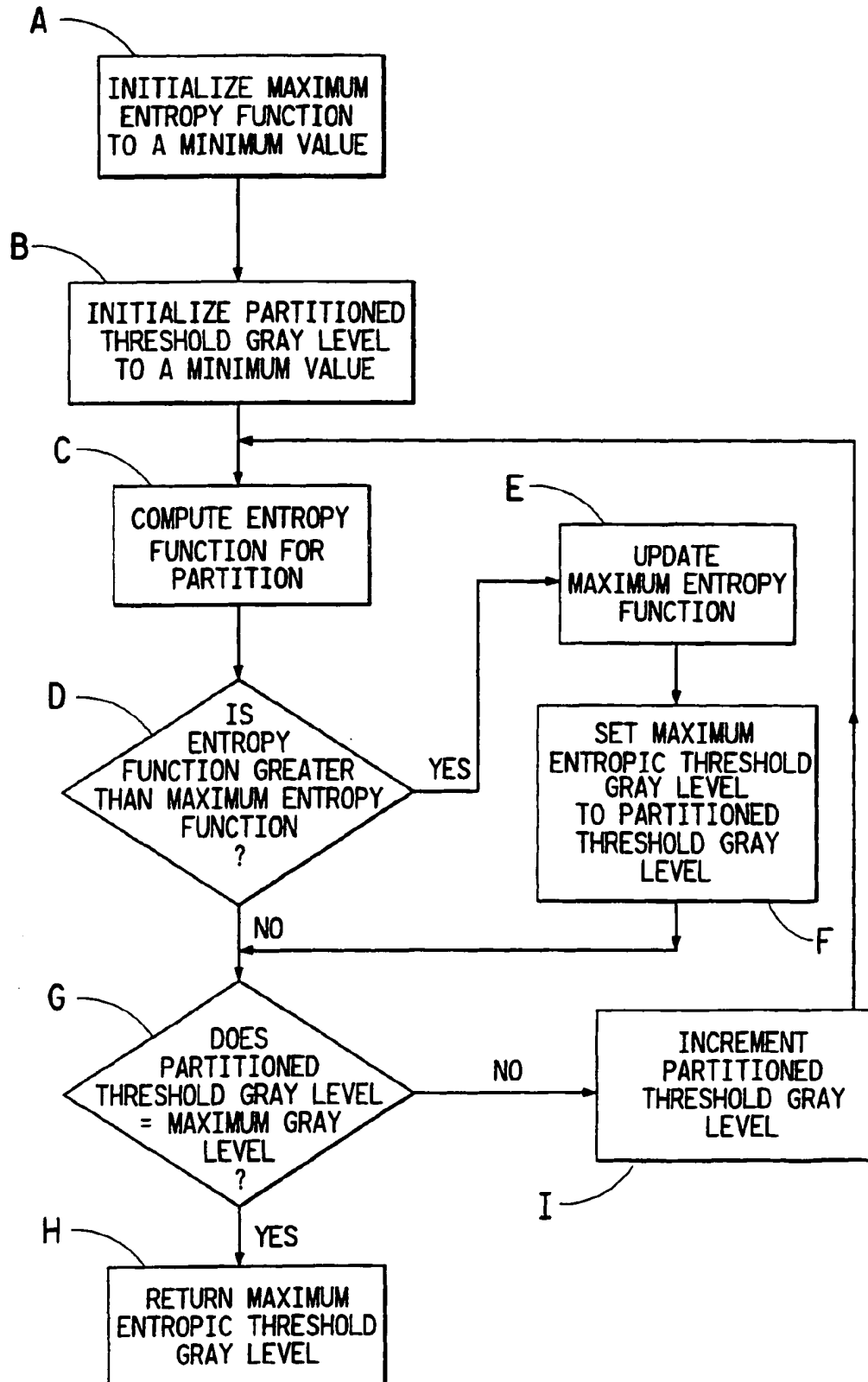


FIG. 5

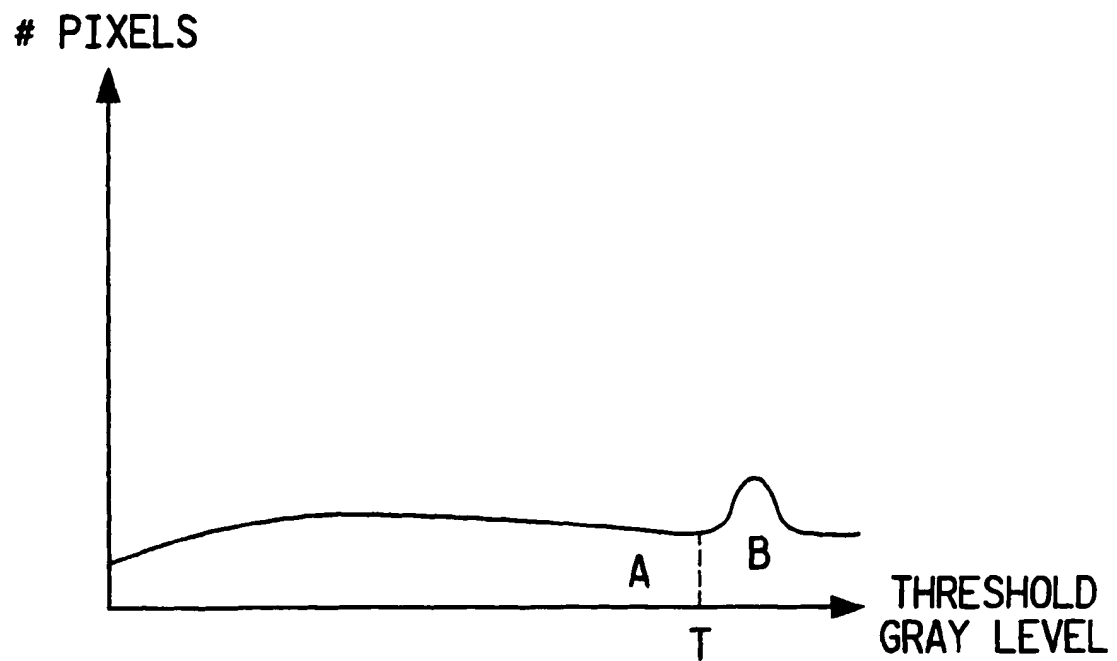


FIG. 6

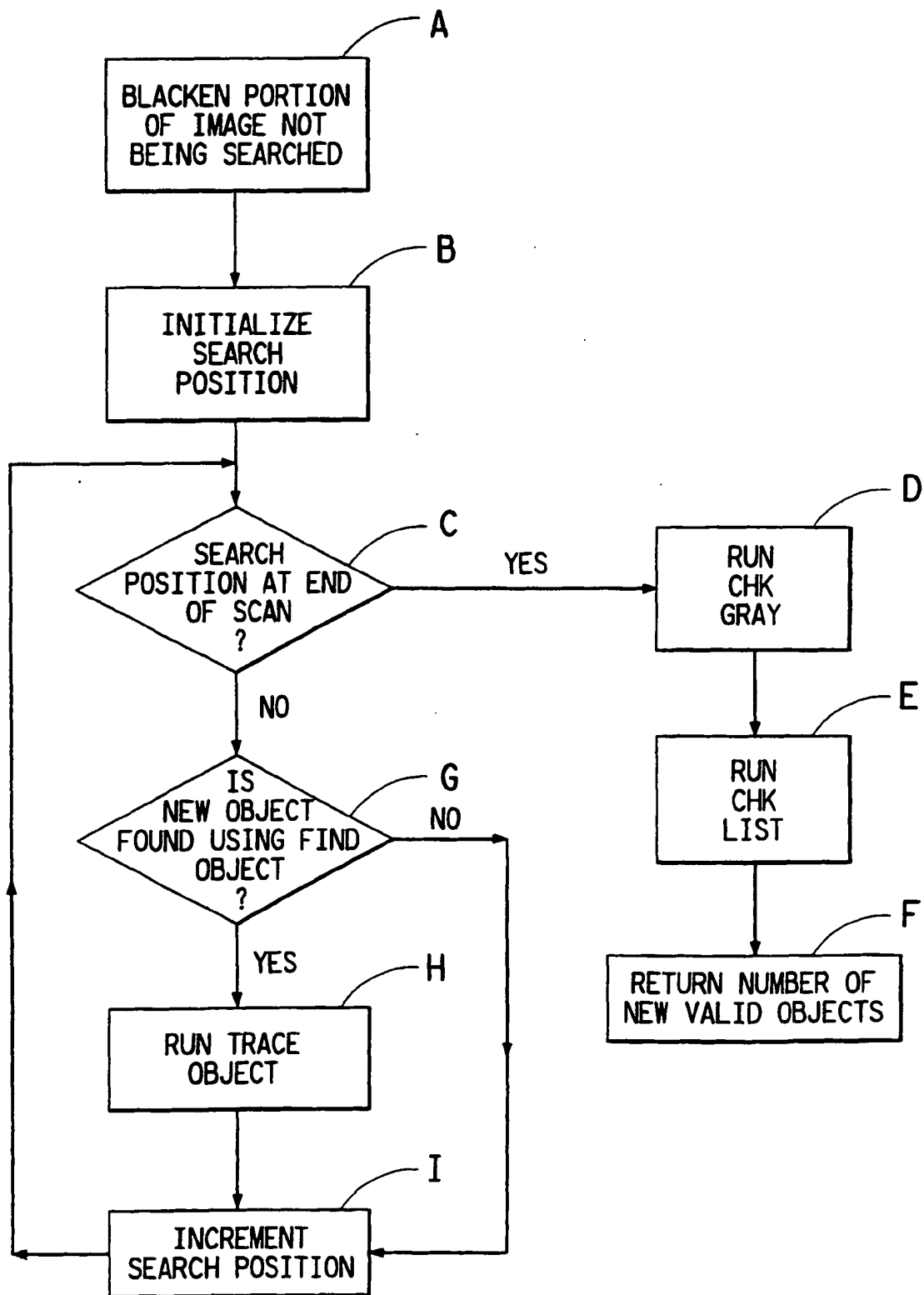


FIG. 7

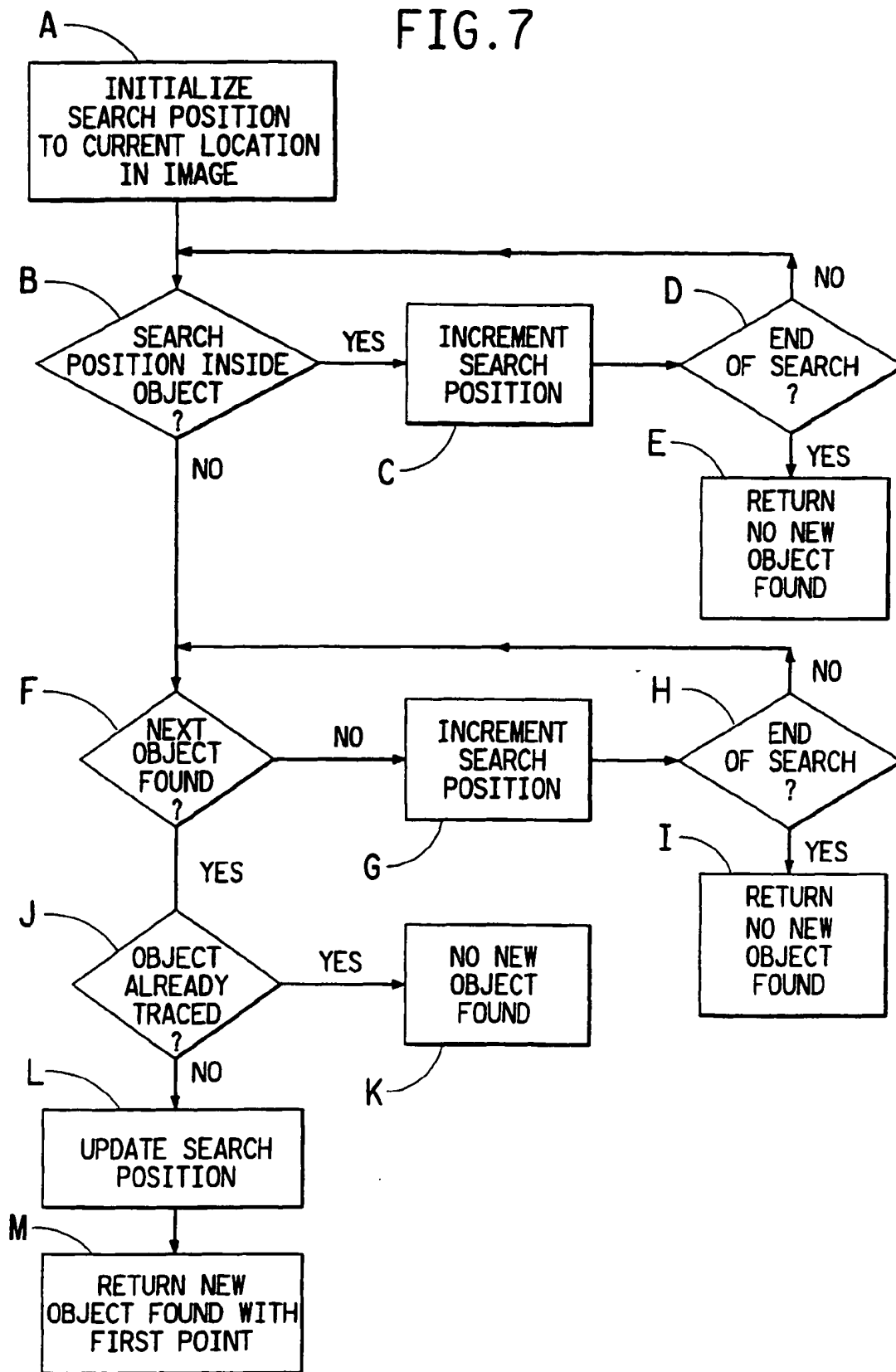


FIG. 8A

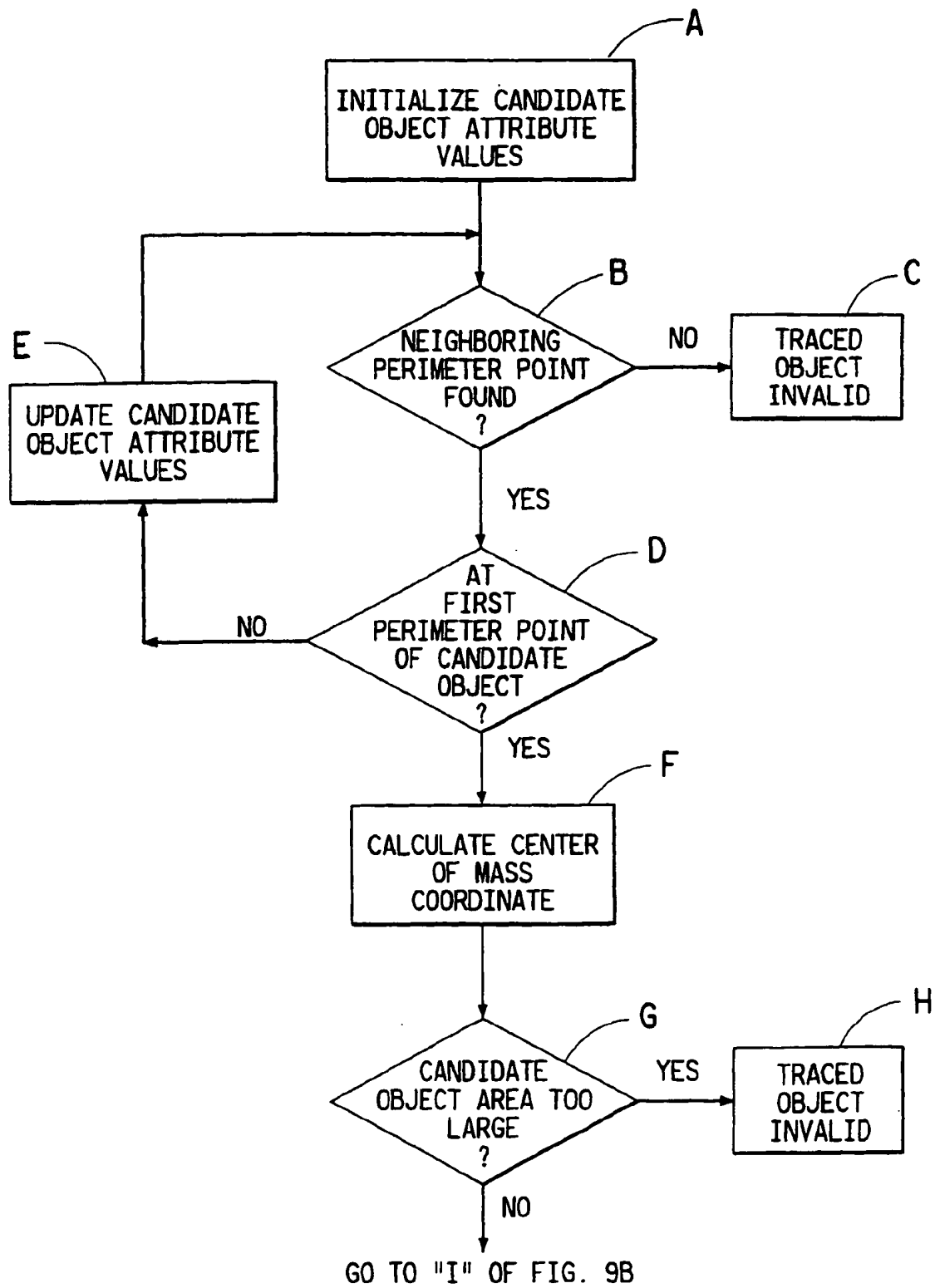


FIG. 8B

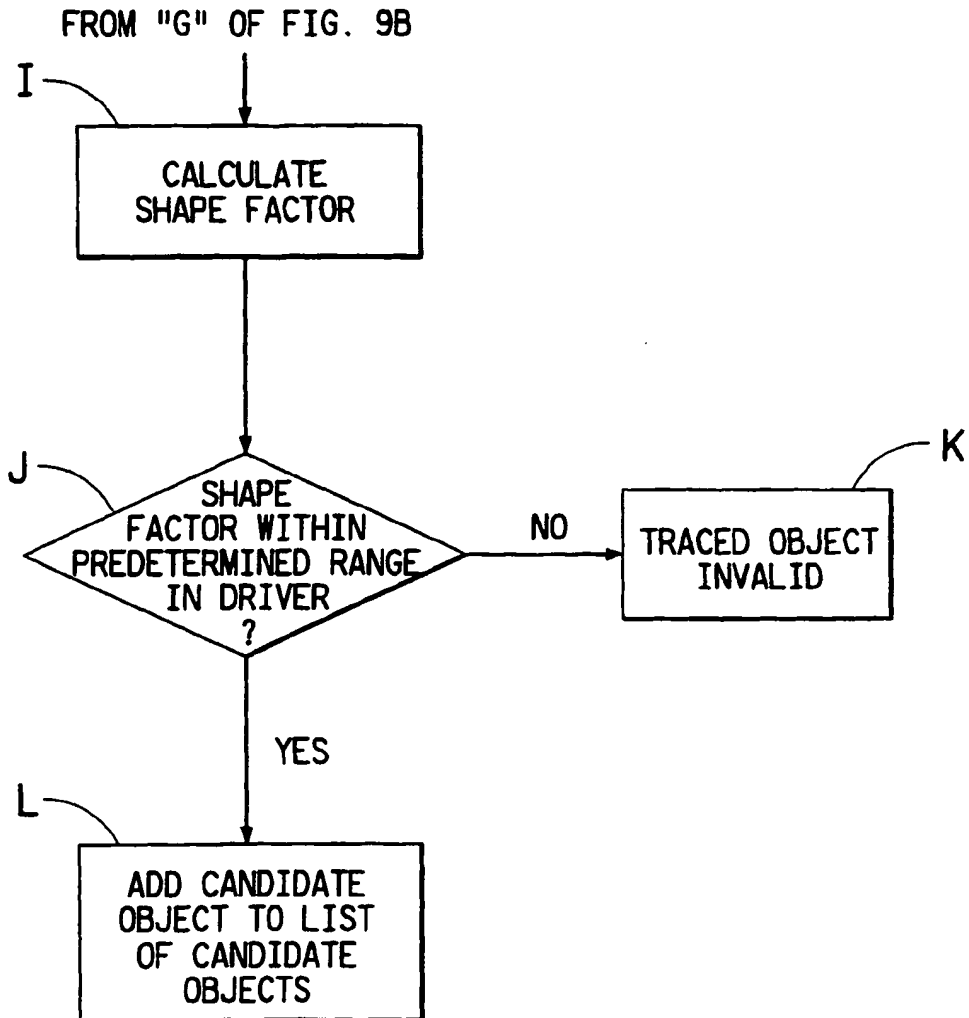


FIG.9

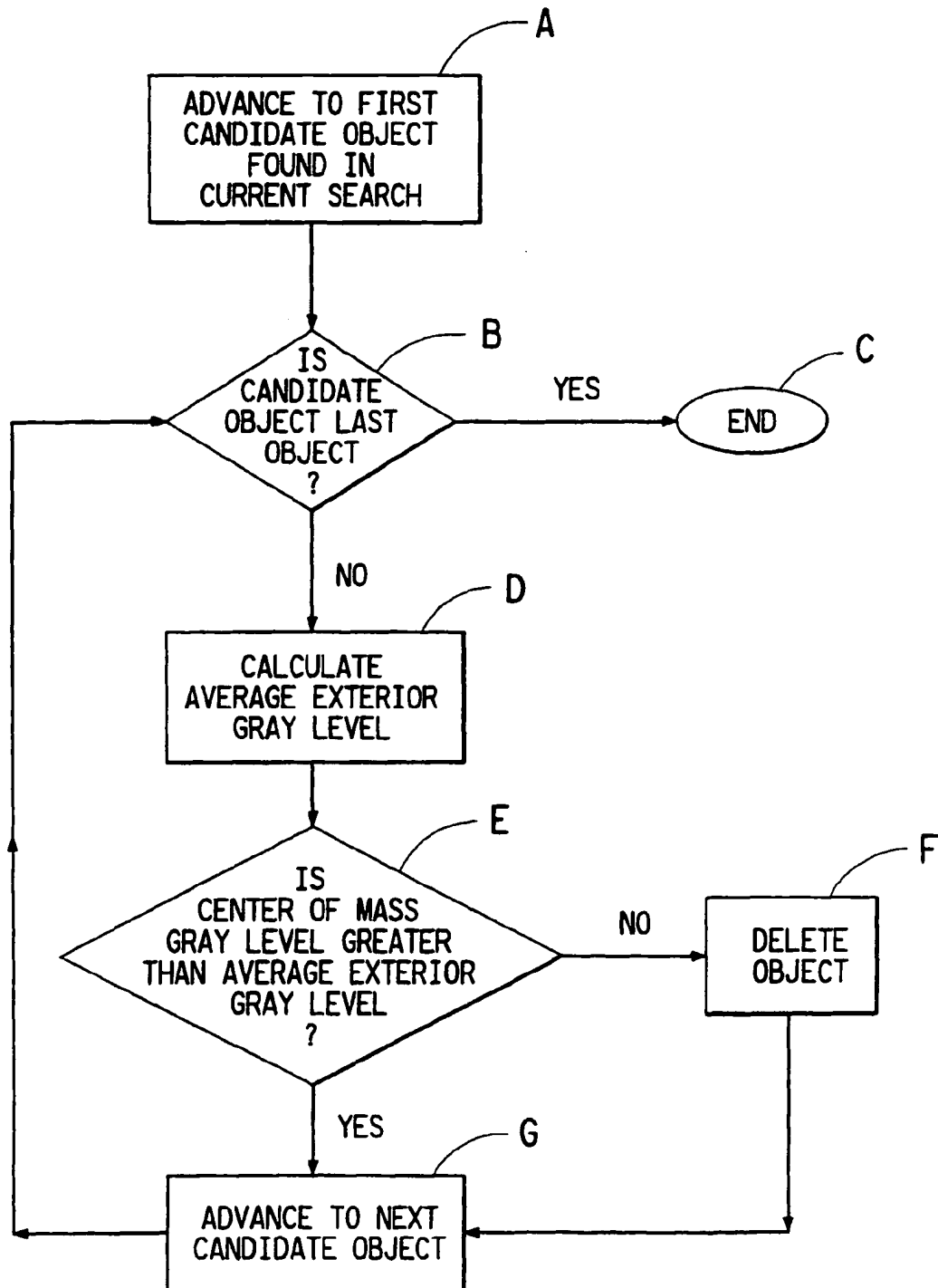


FIG. 10A

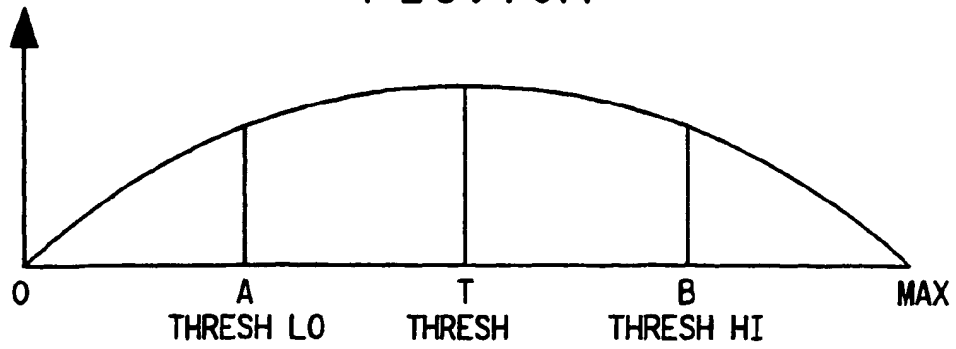


FIG. 10B

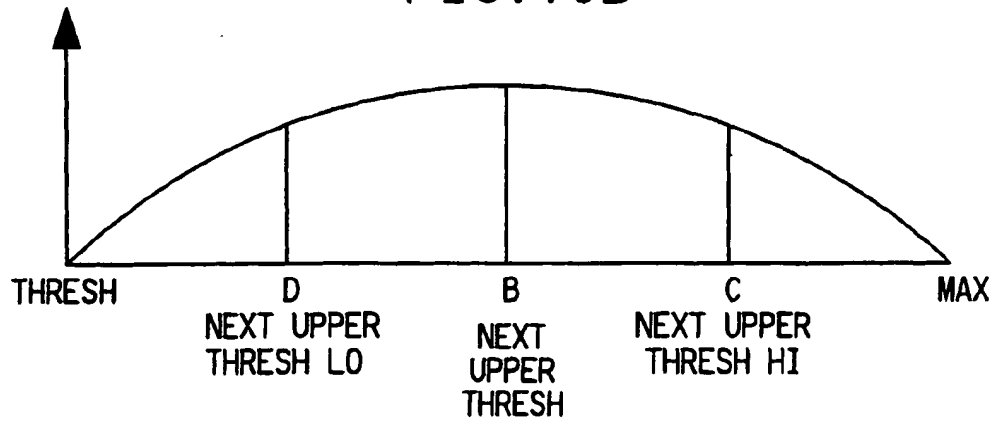


FIG. 10C

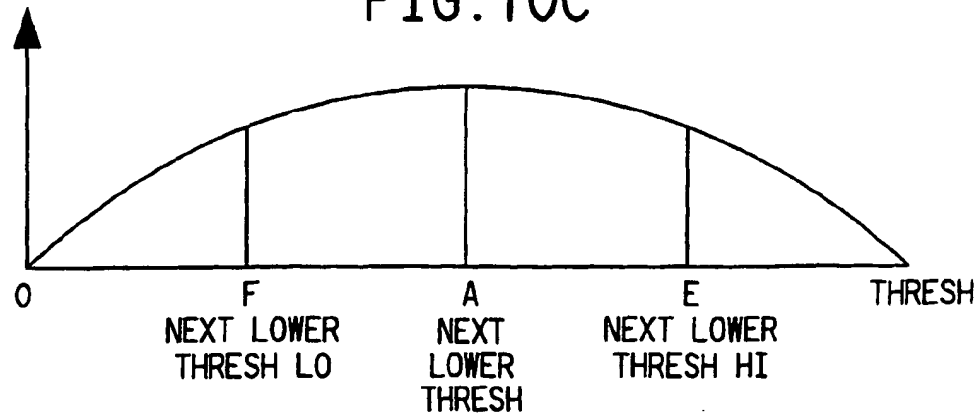


FIG. 11

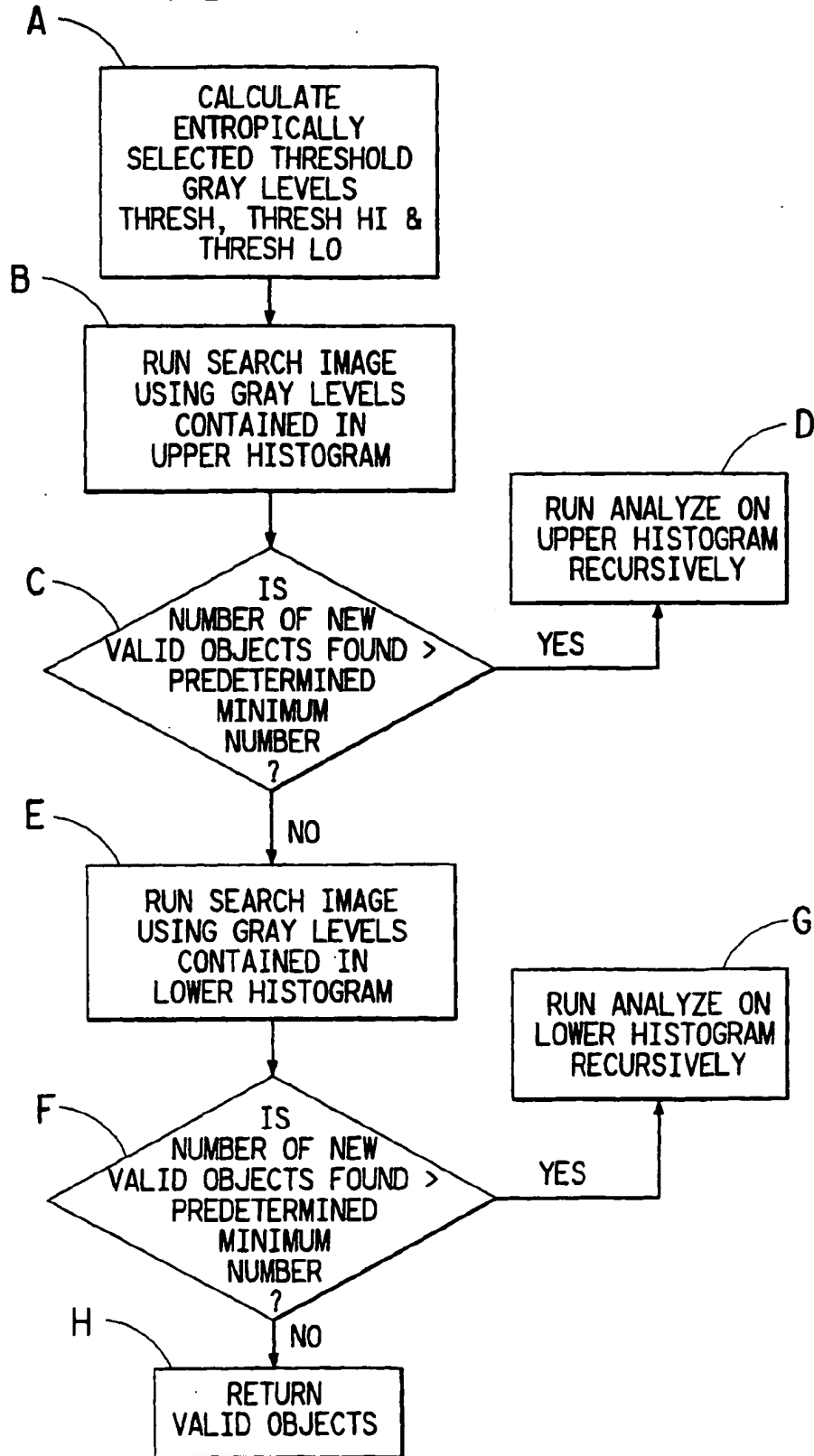


FIG. 12

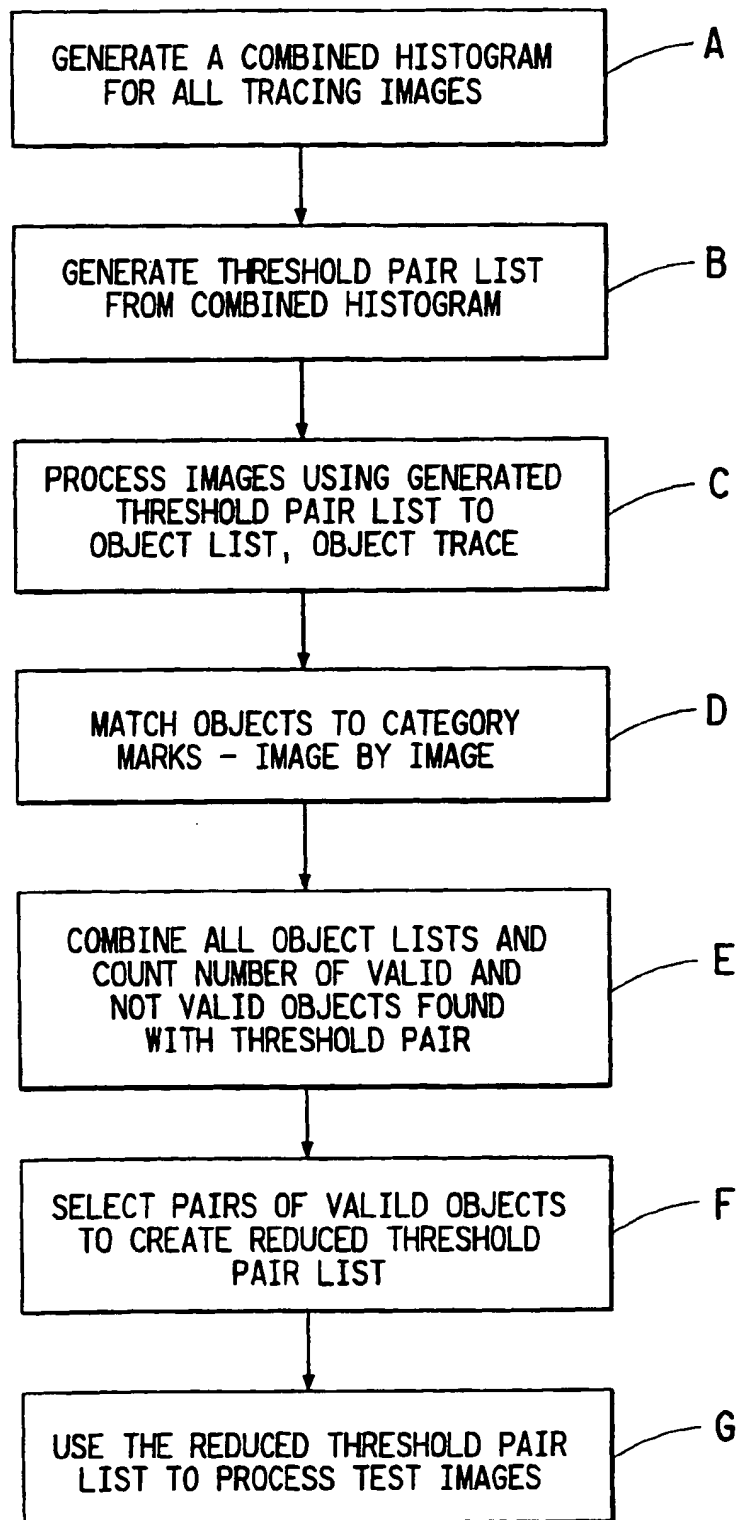


FIG. 13A

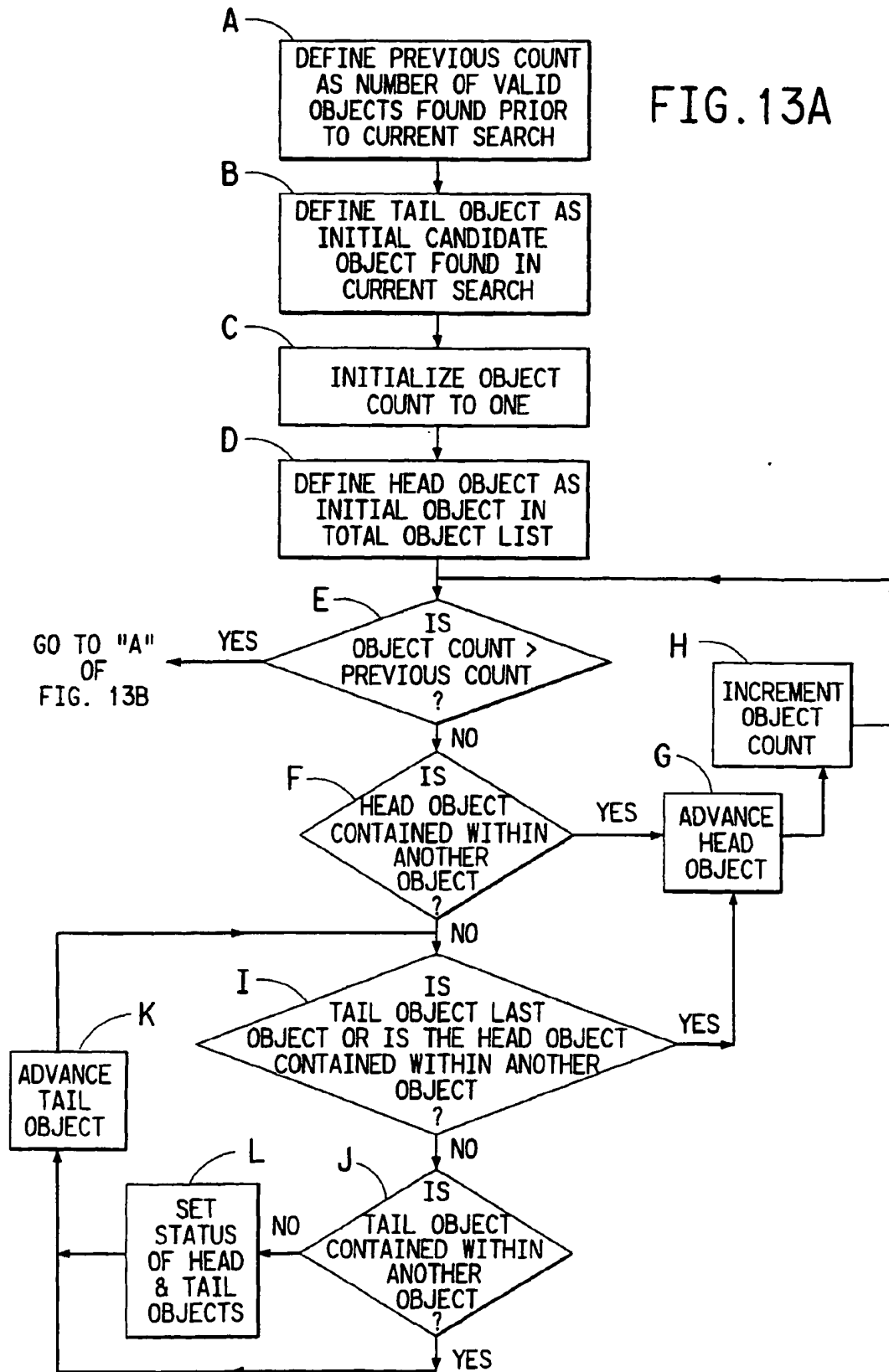


FIG. 13B

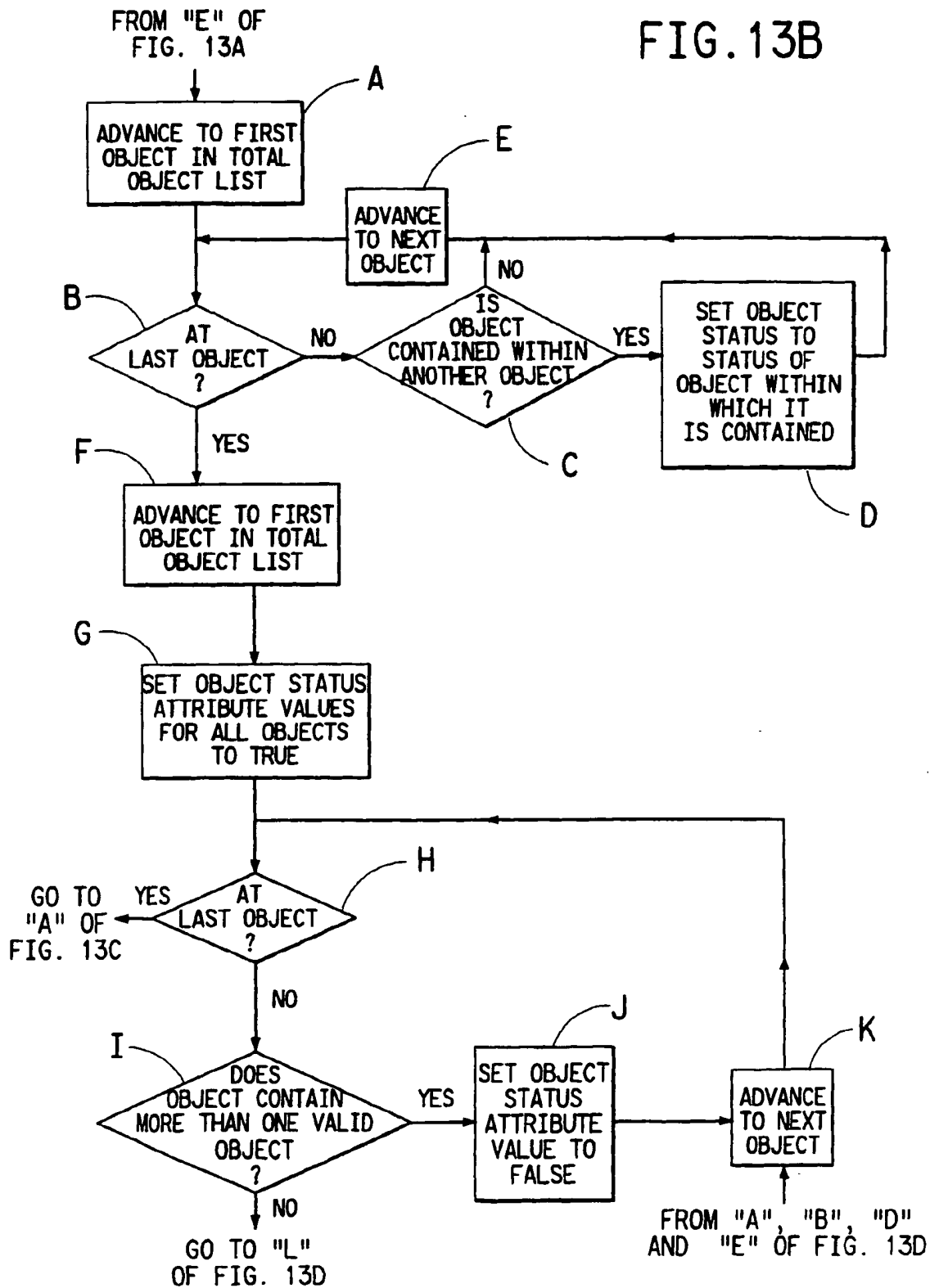


FIG. 13C

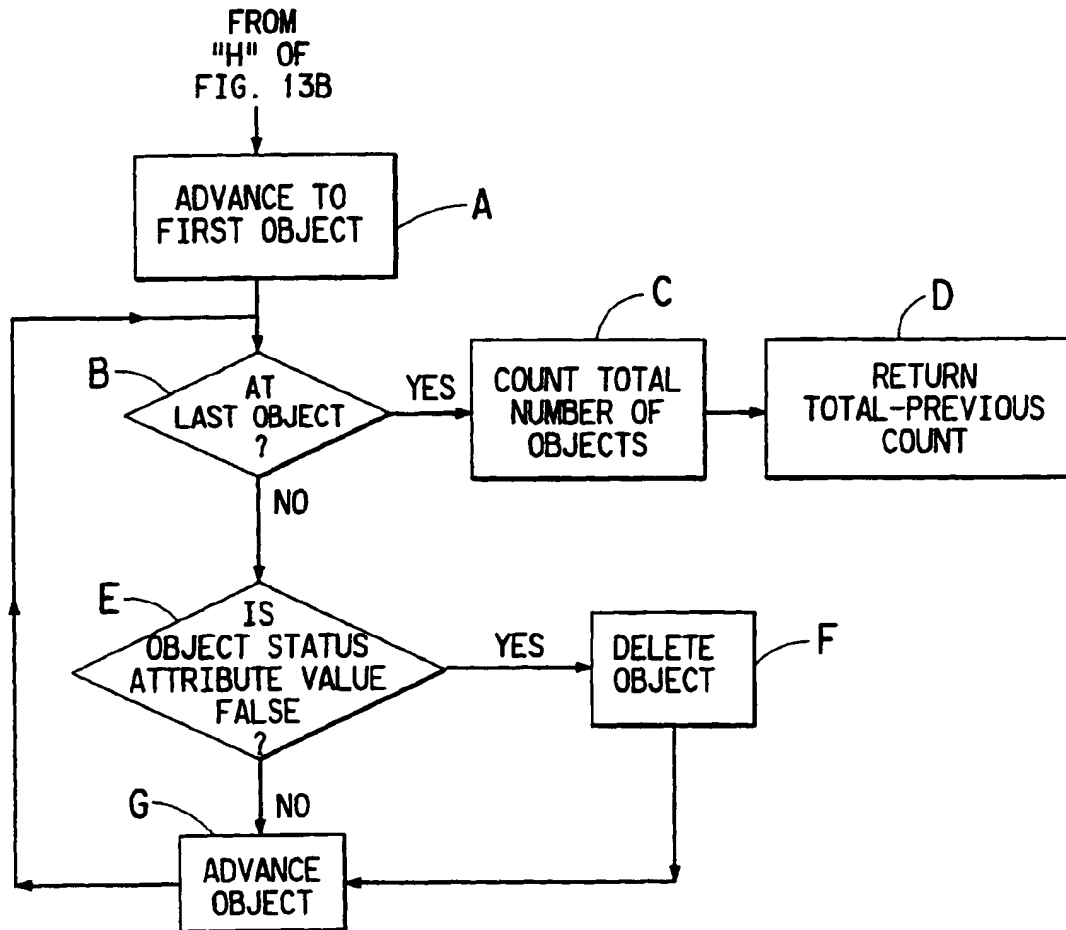


FIG. 13D

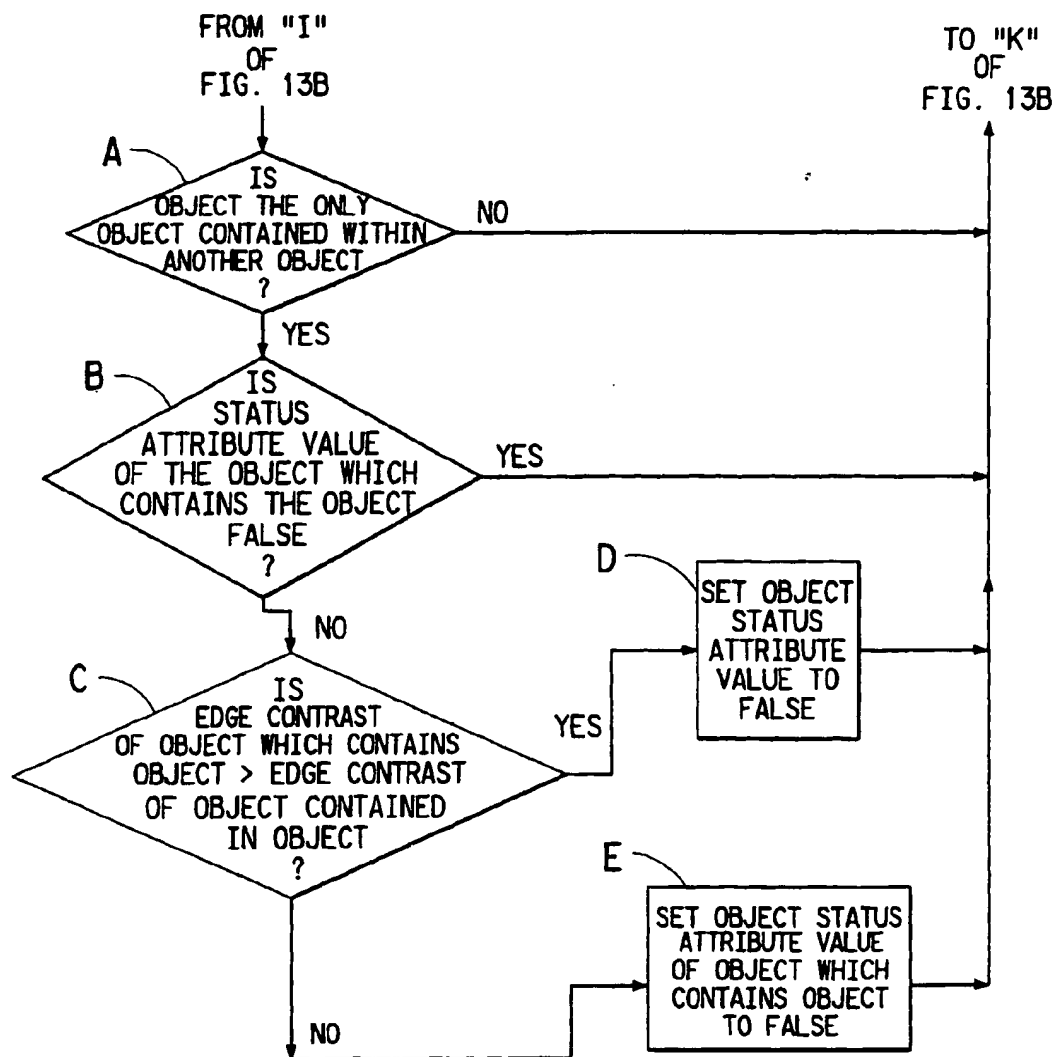


FIG. 14

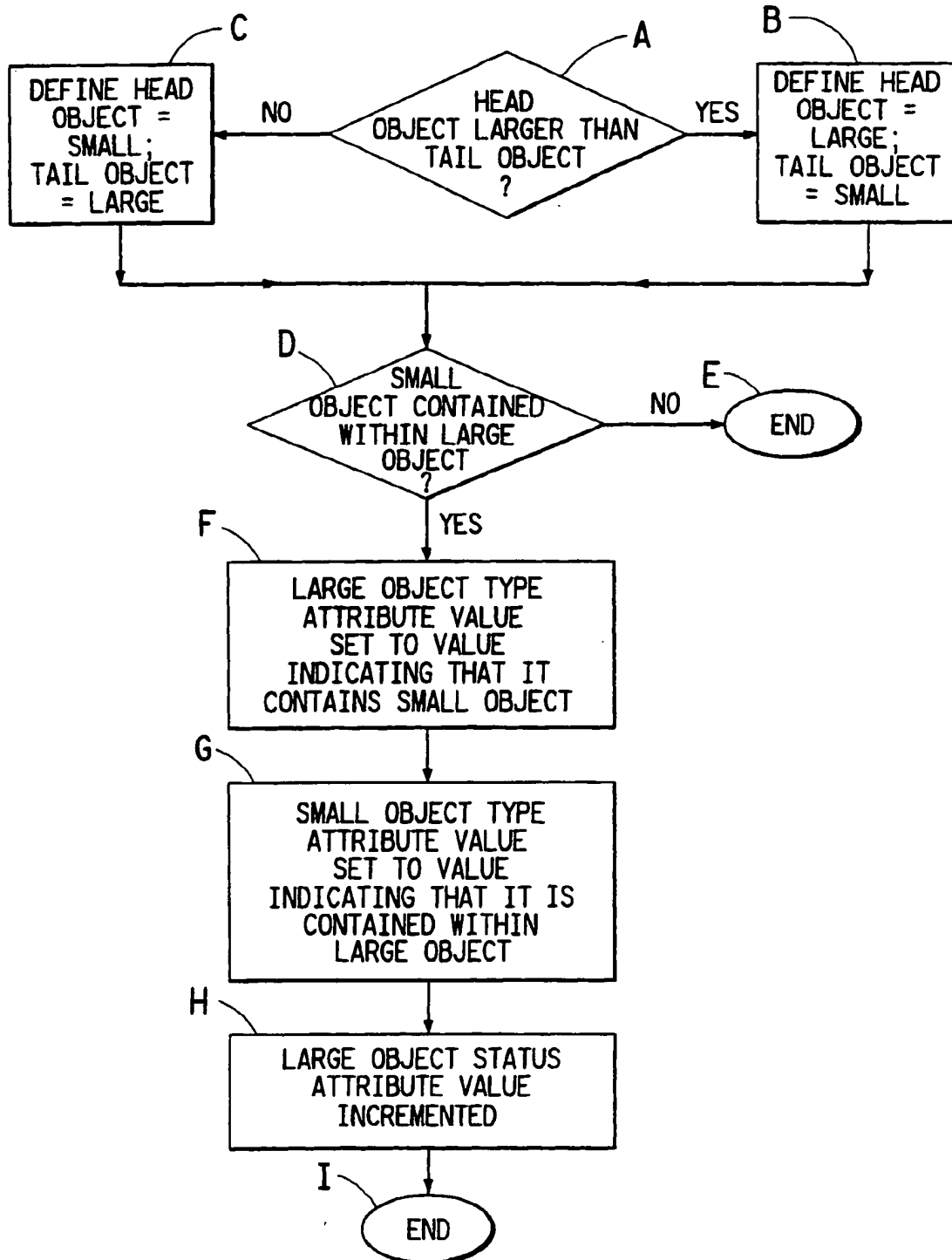


FIG. 15A

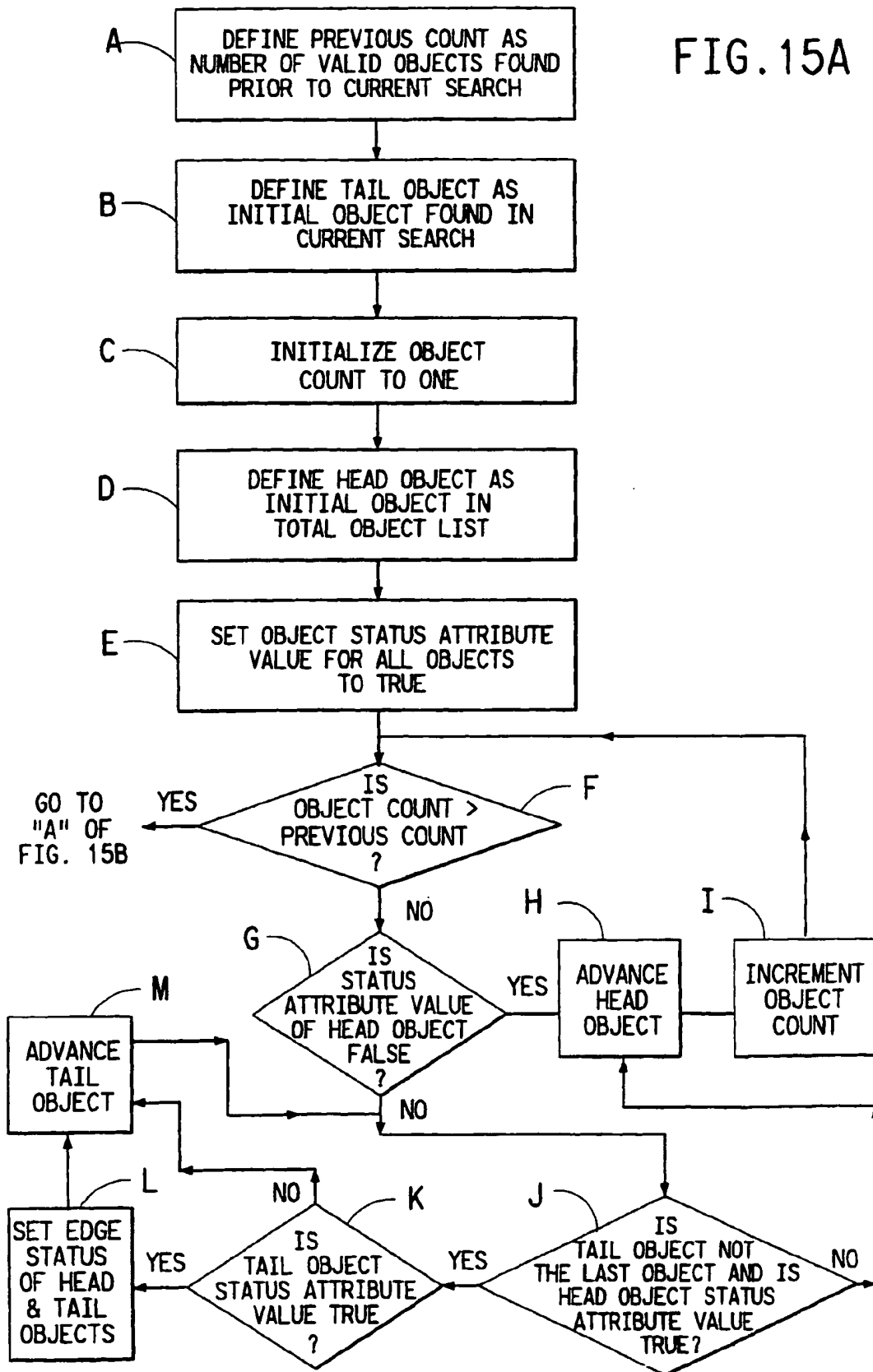


FIG. 15B

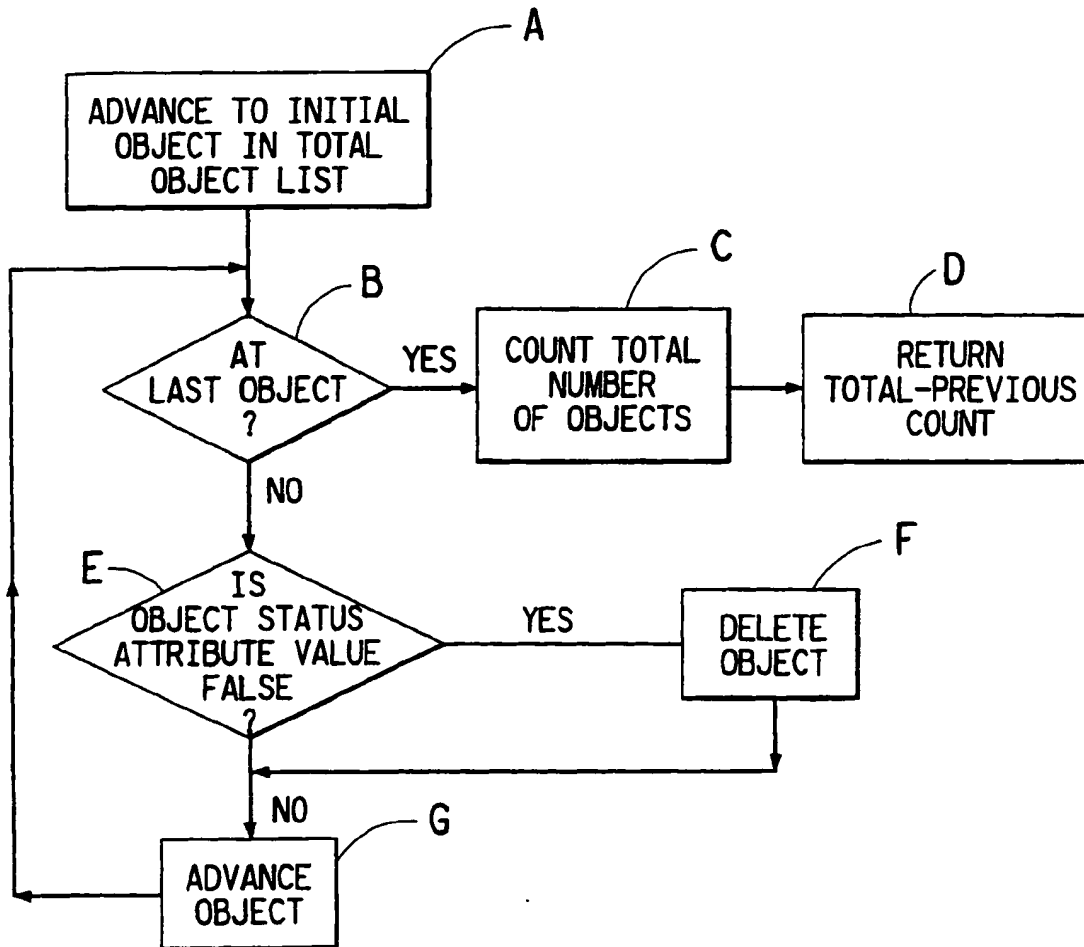


FIG. 16

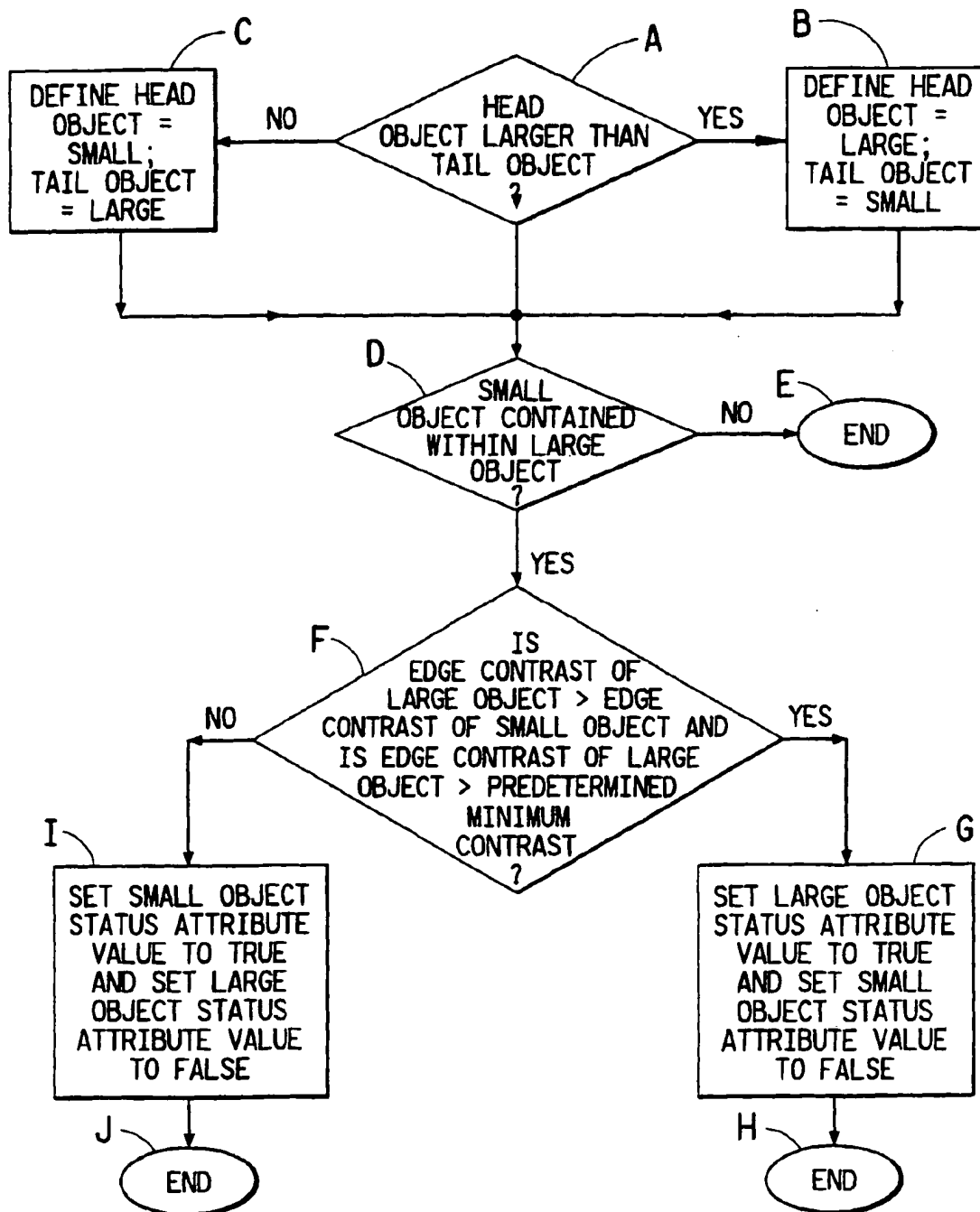


FIG. 17A

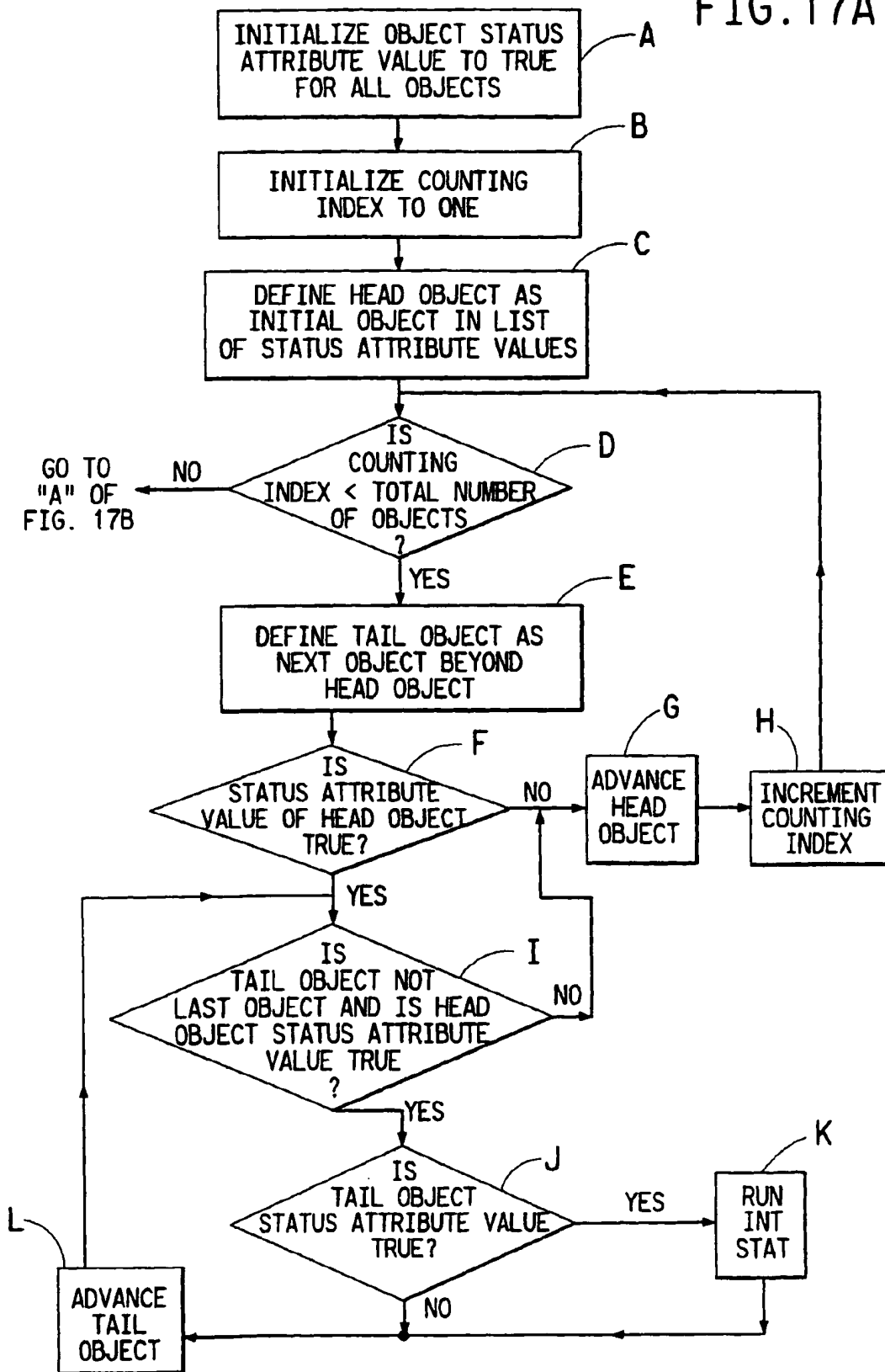


FIG. 17B

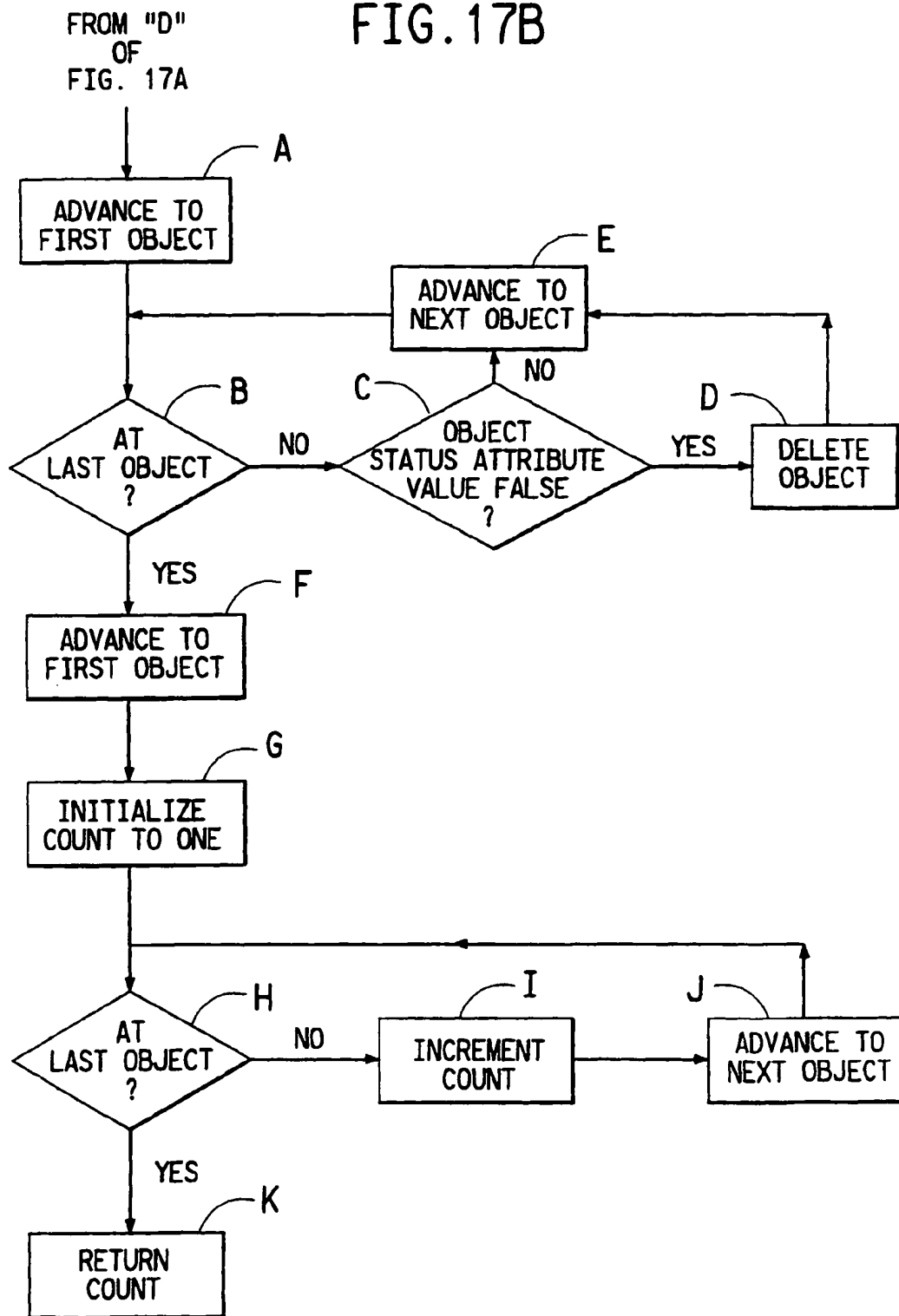


FIG. 18

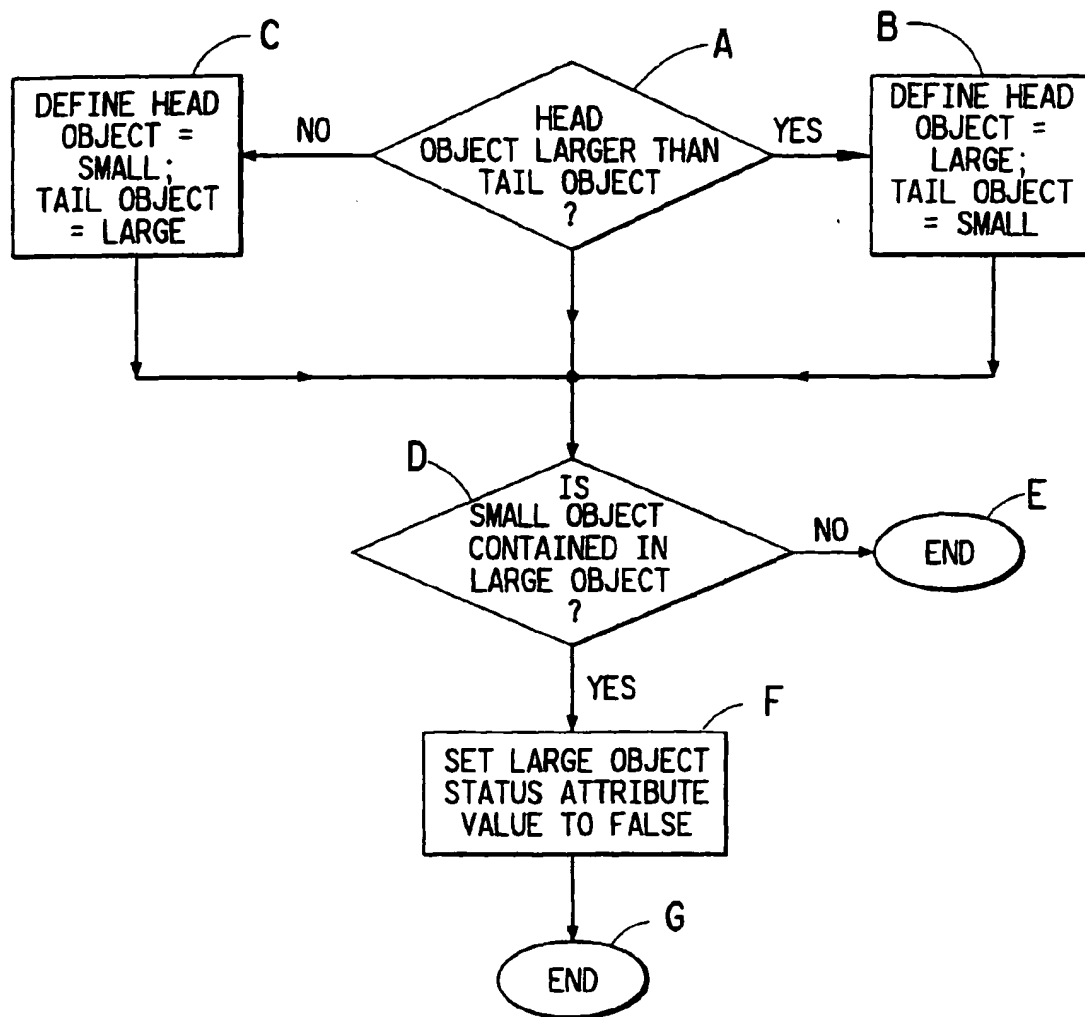


FIG. 19A

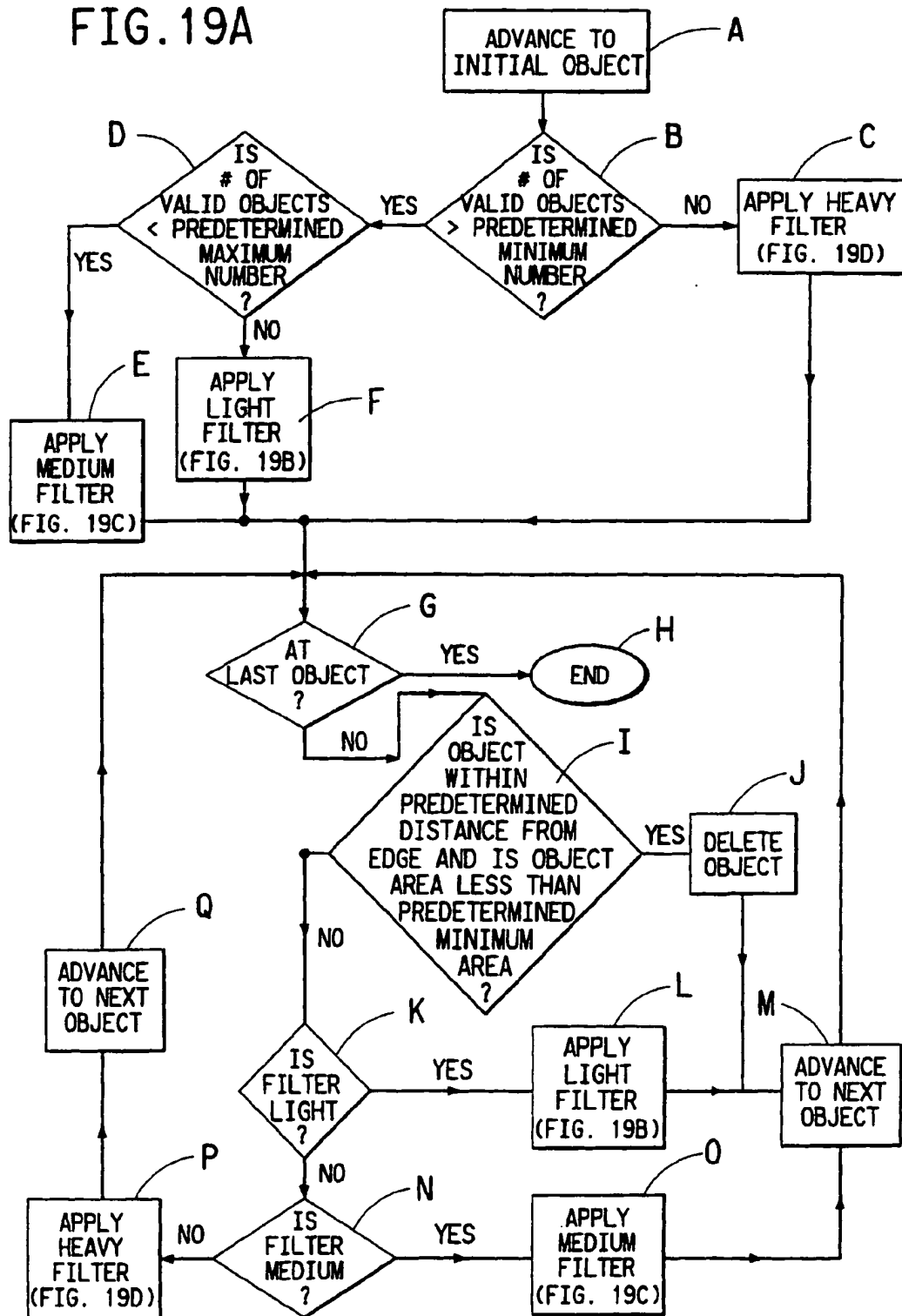


FIG. 19B

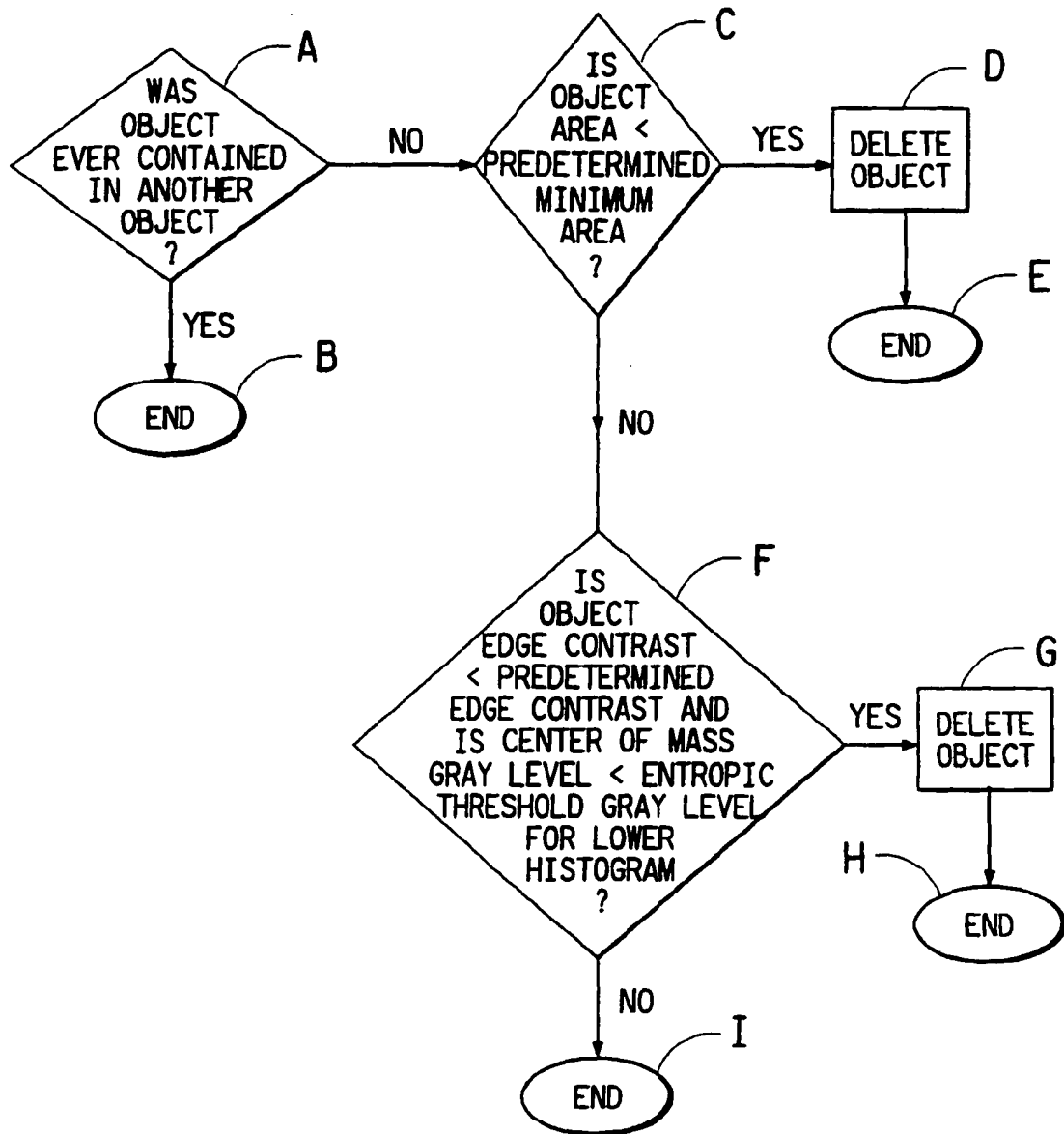


FIG. 19C

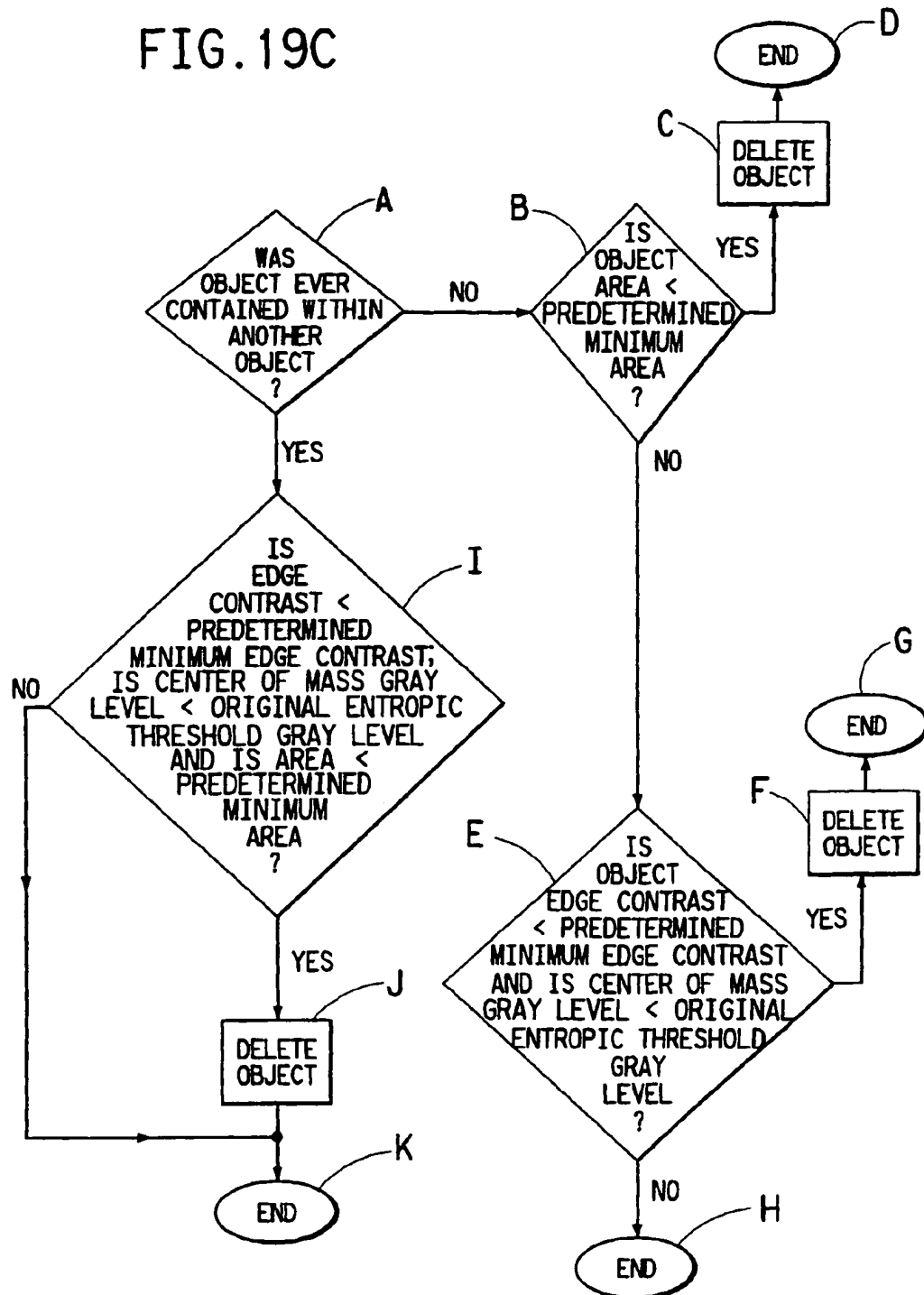
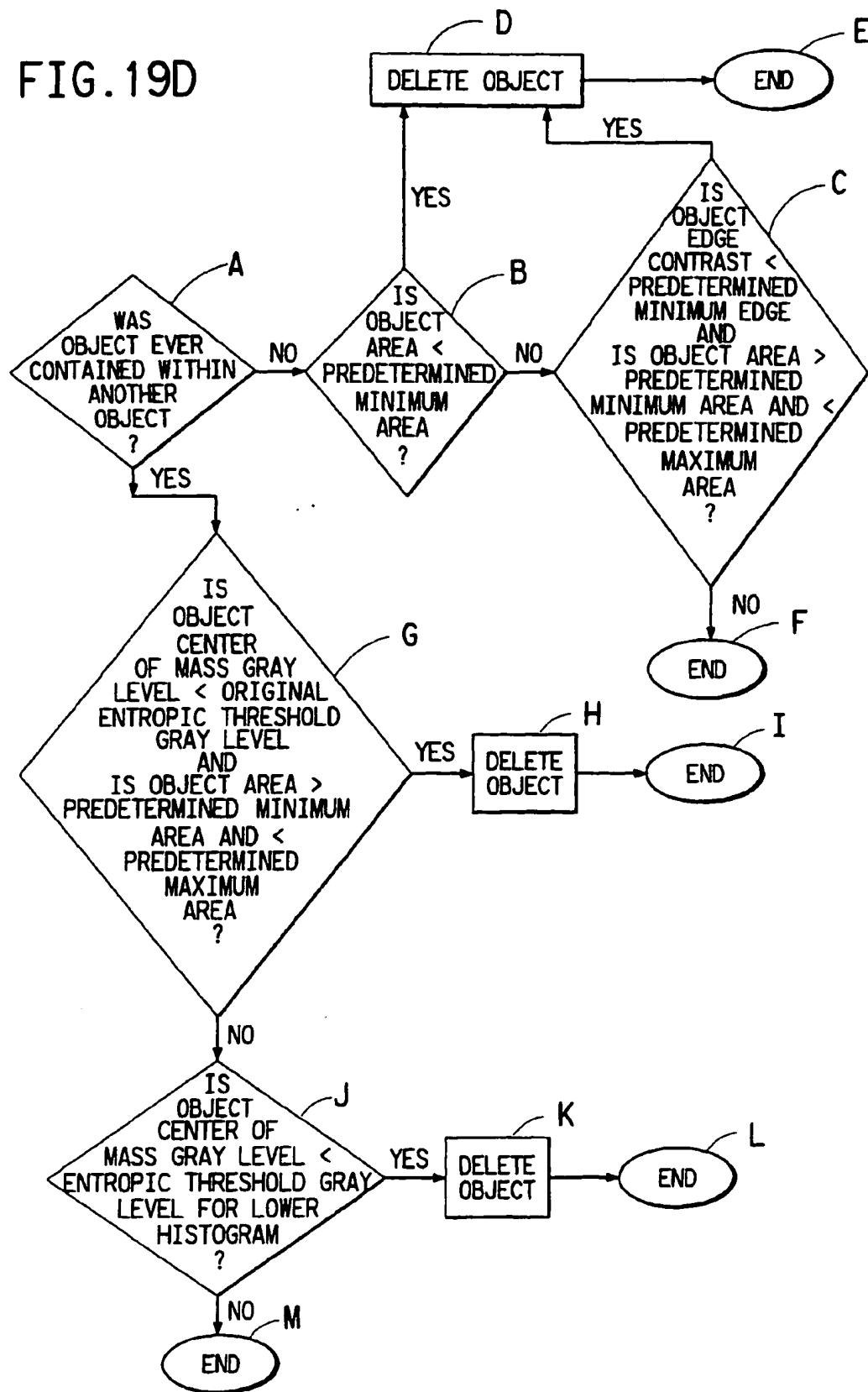


FIG. 19D



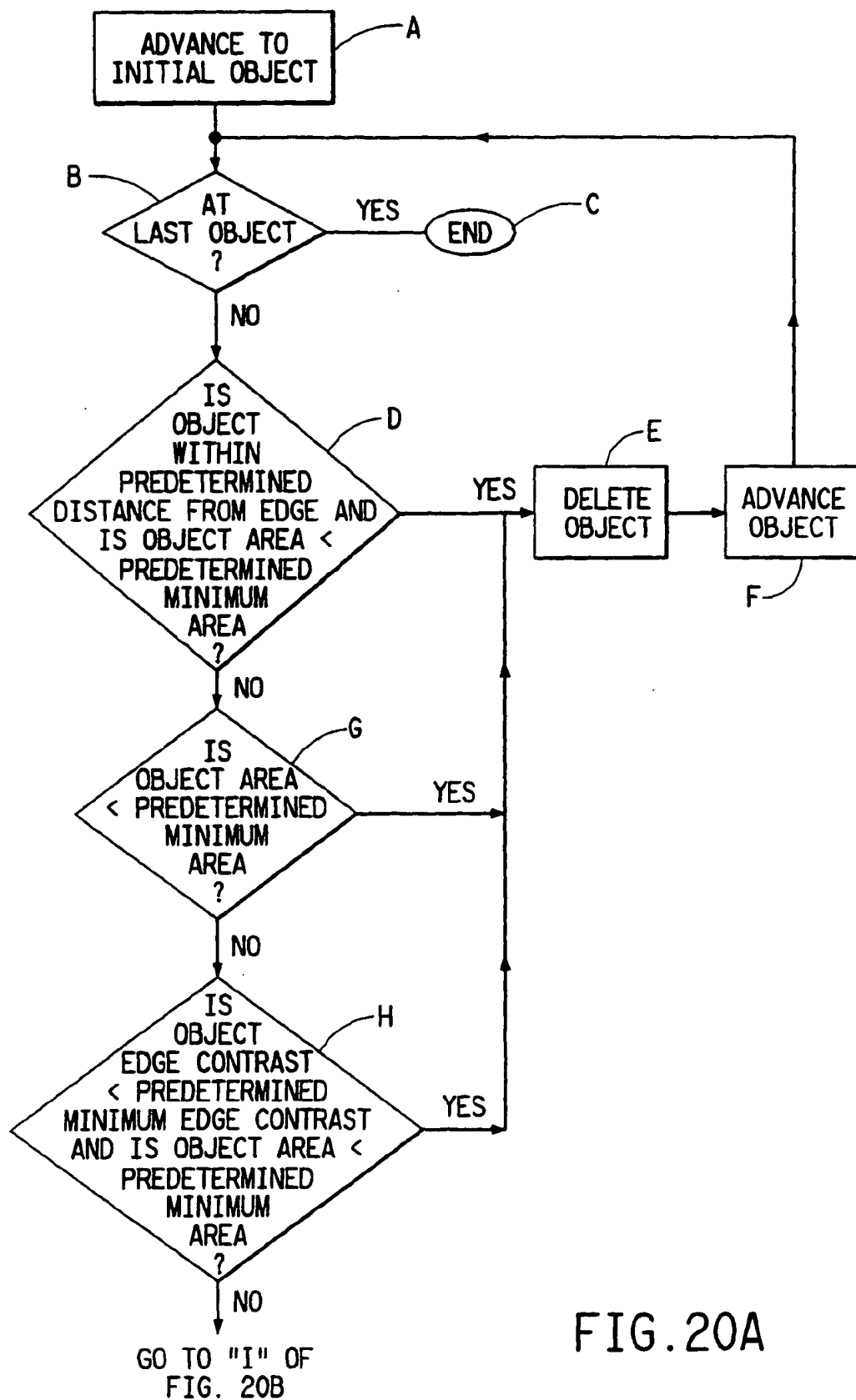


FIG. 20A

FIG. 20B

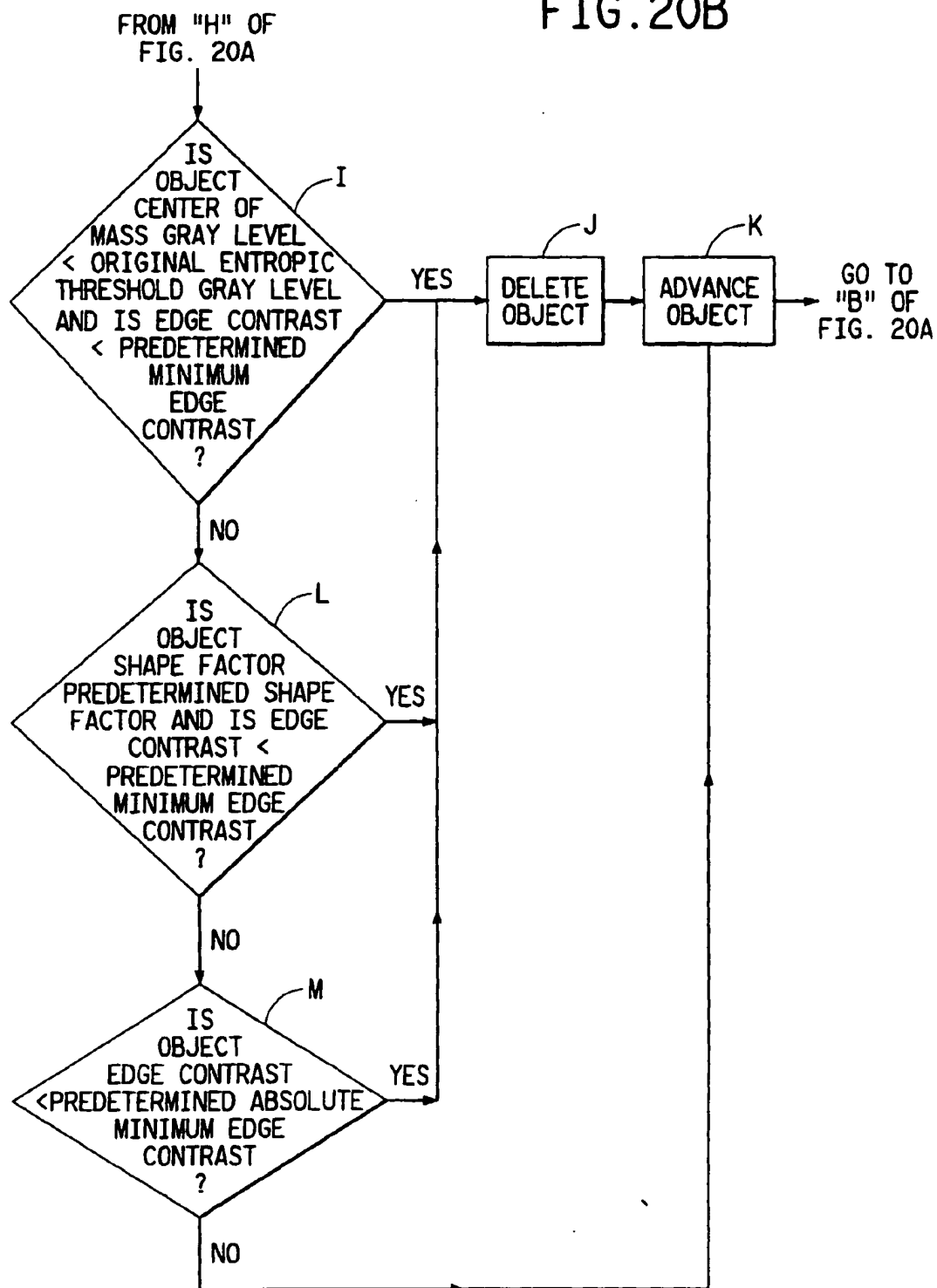


FIG. 21A

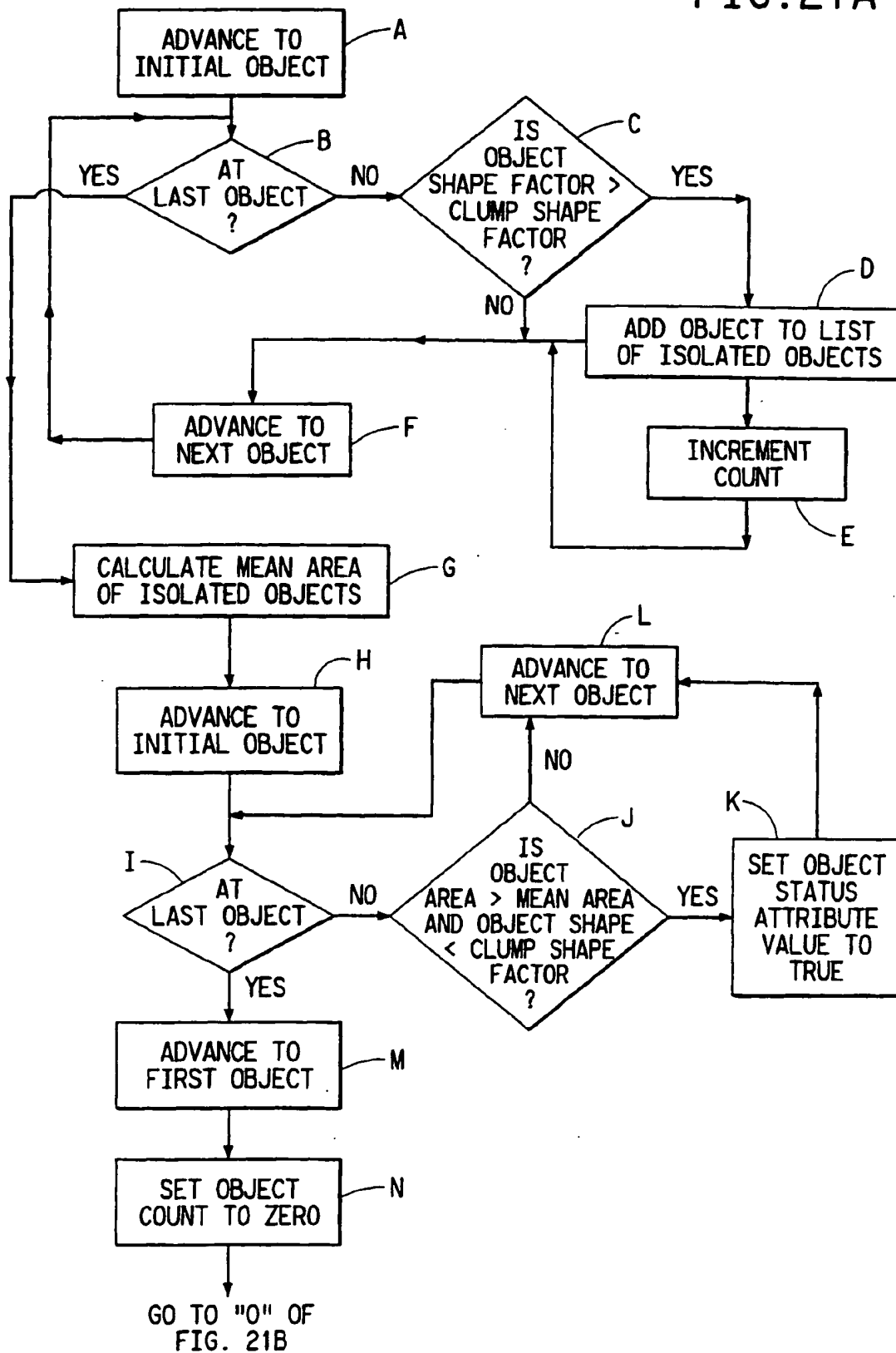


FIG. 21B

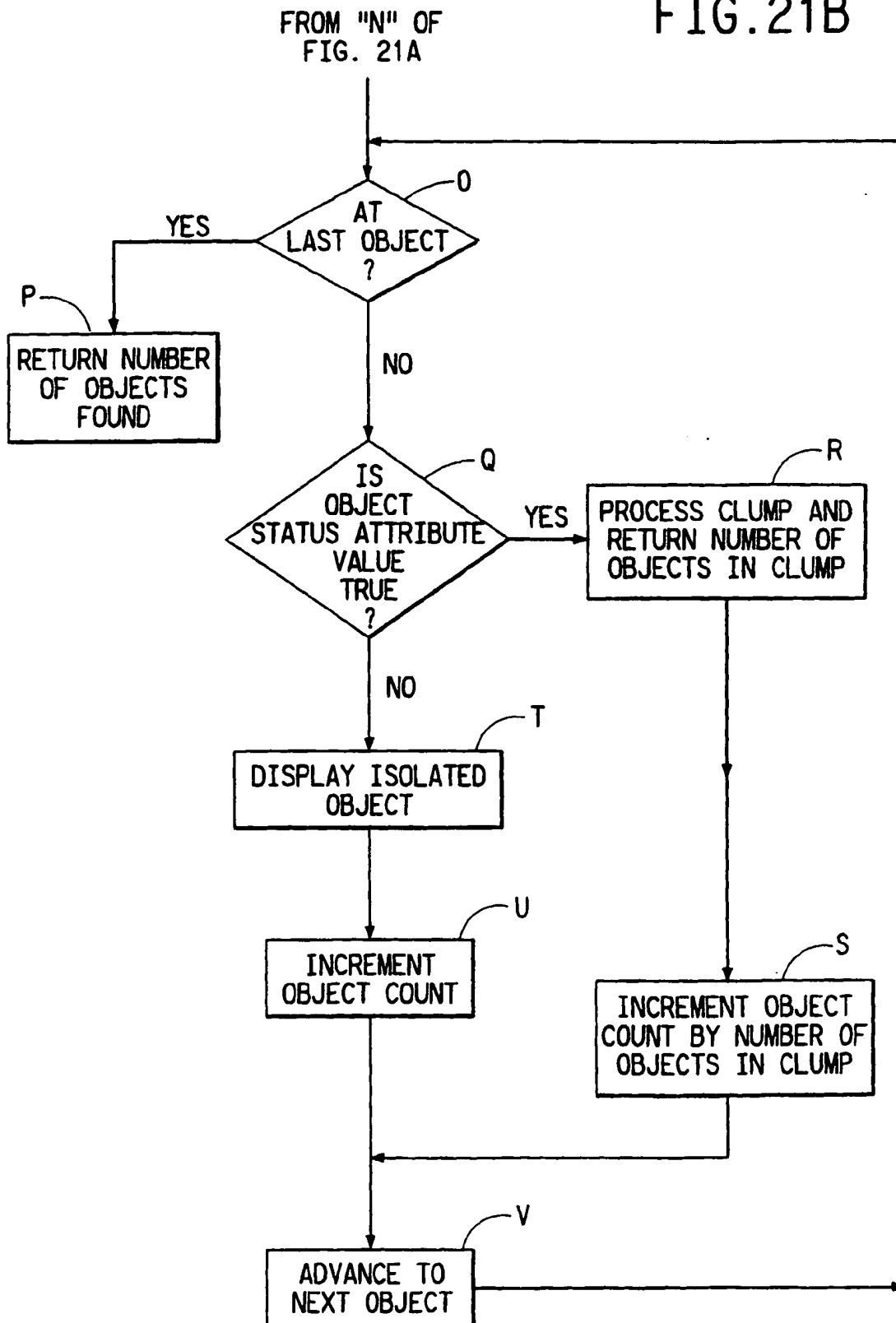


FIG. 22A

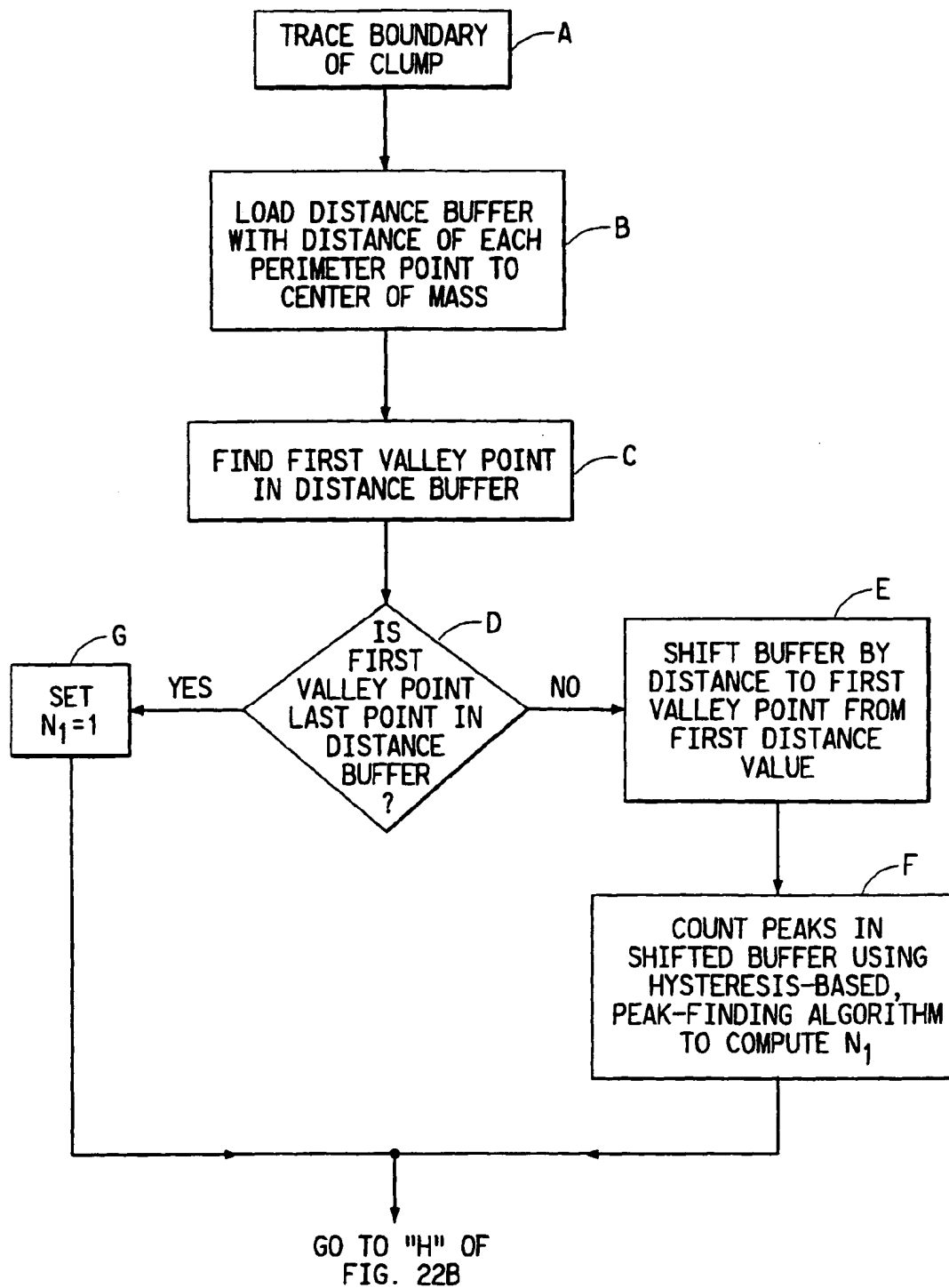


FIG. 22B

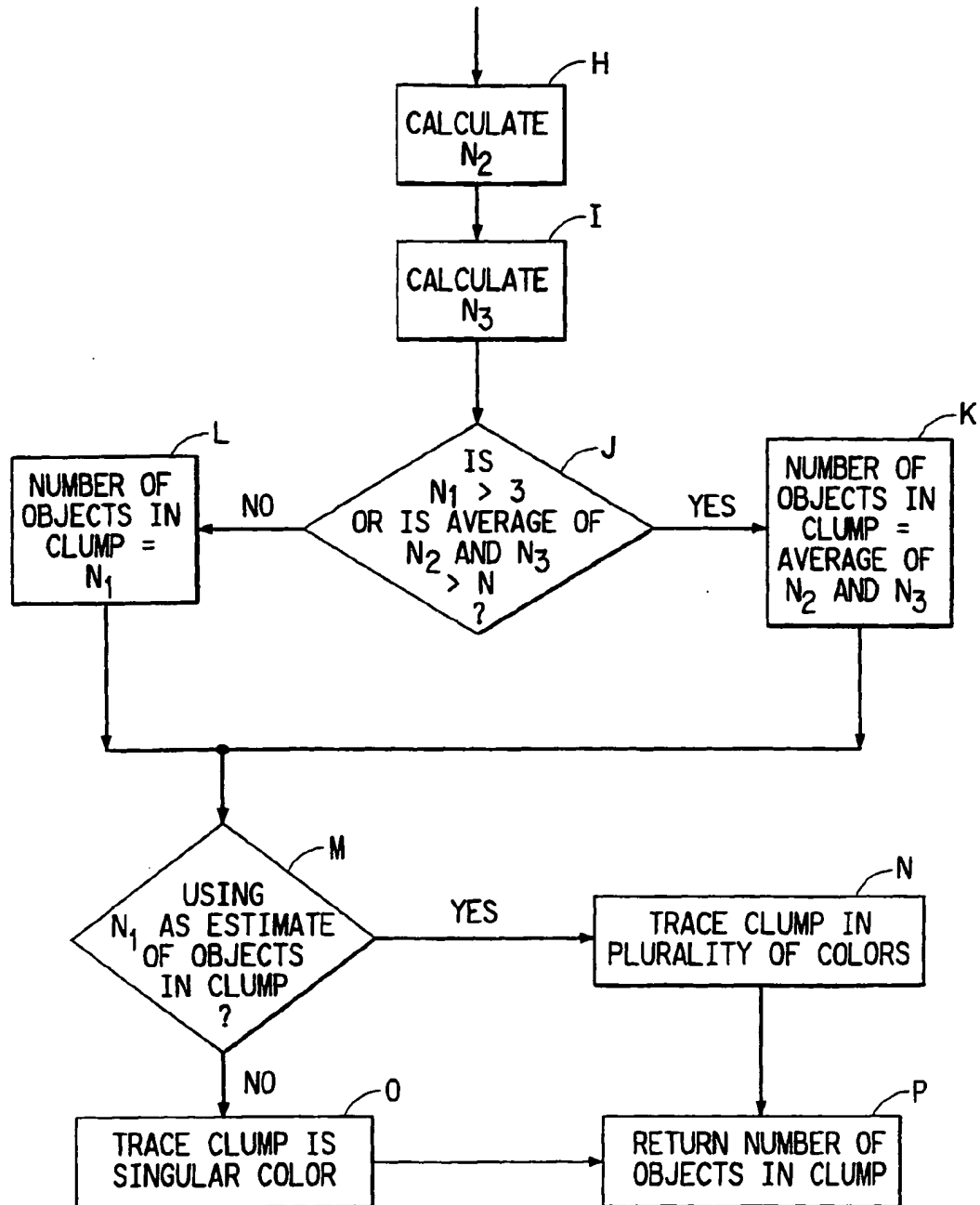
FROM "G" OR "F"
OF FIG. 22A

FIG. 23A

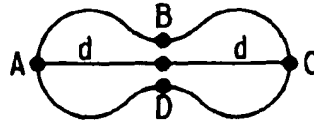


FIG. 23B

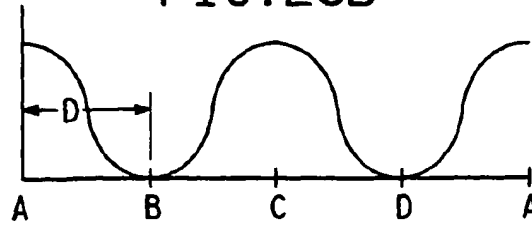


FIG. 23C

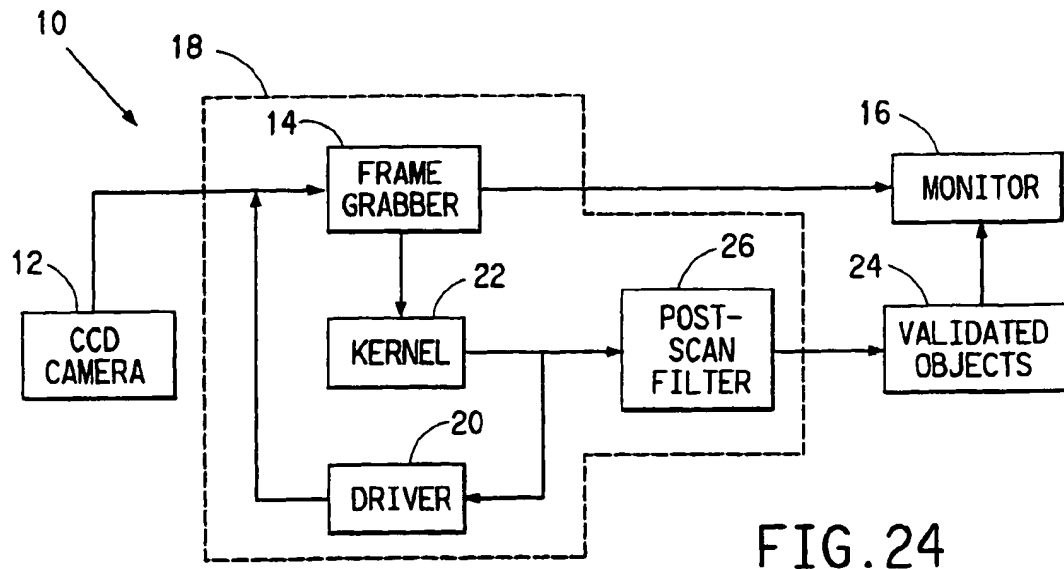
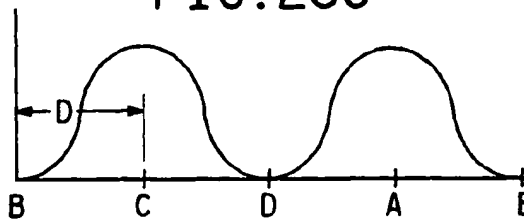


FIG. 24

FIG. 25

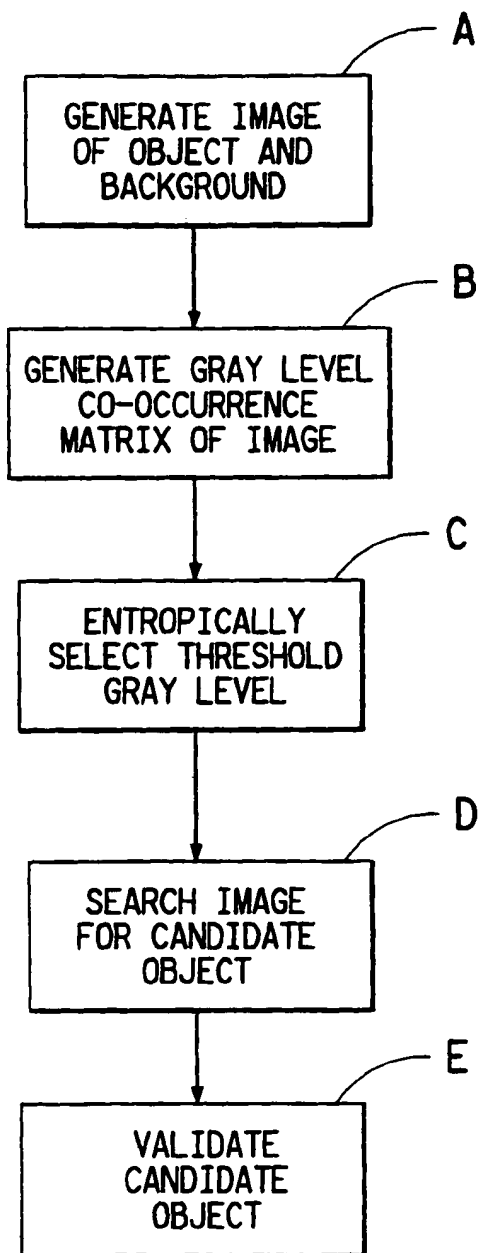


FIG. 26

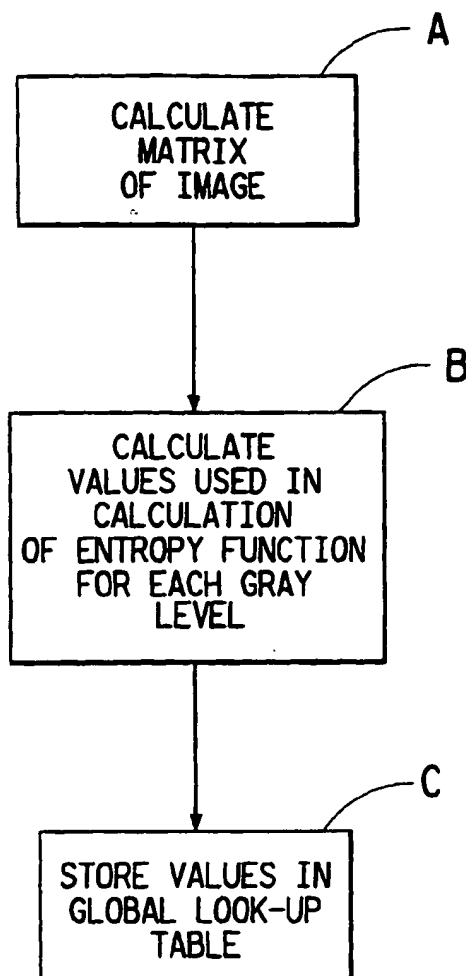


FIG. 27

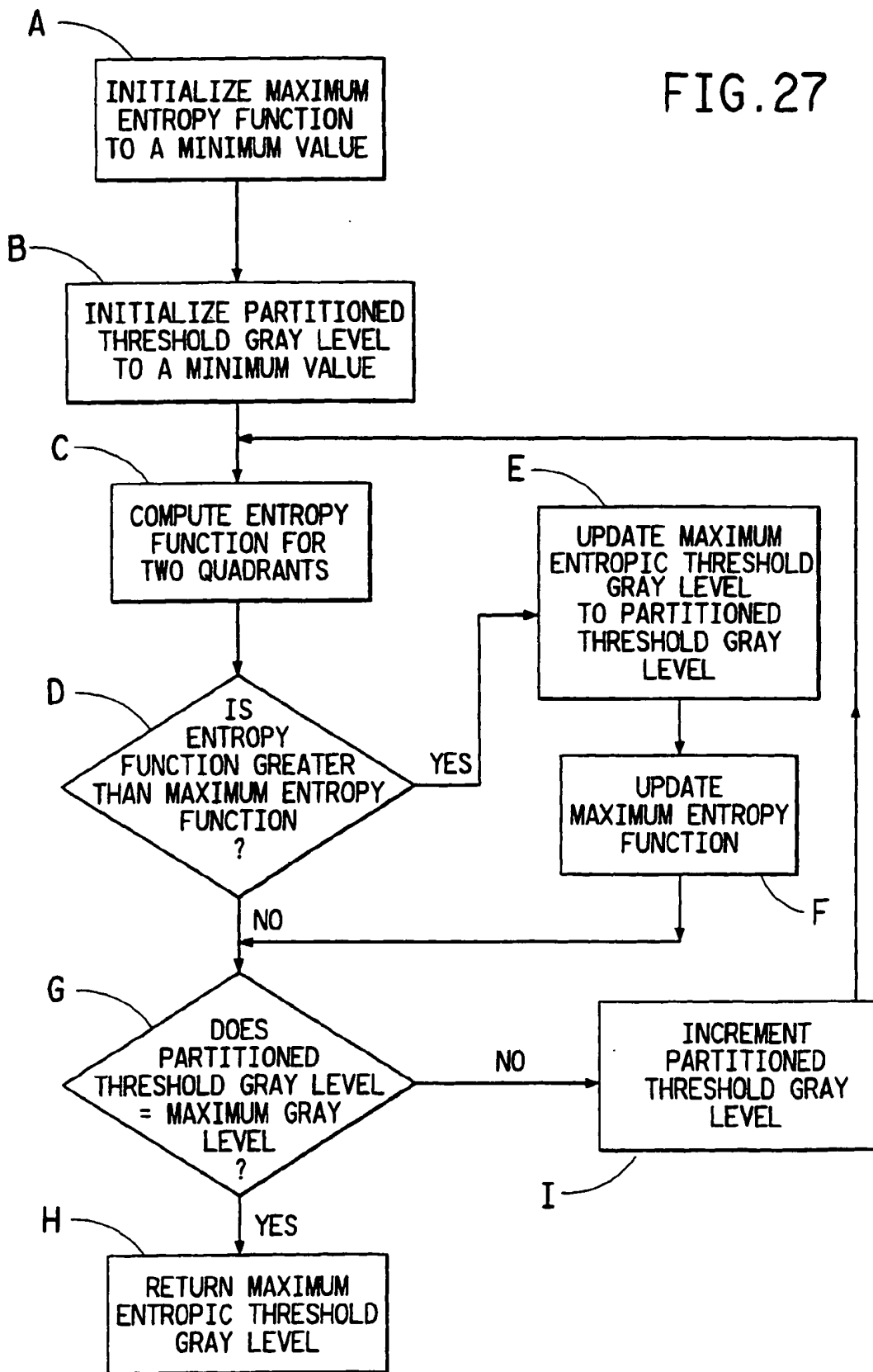


FIG.28

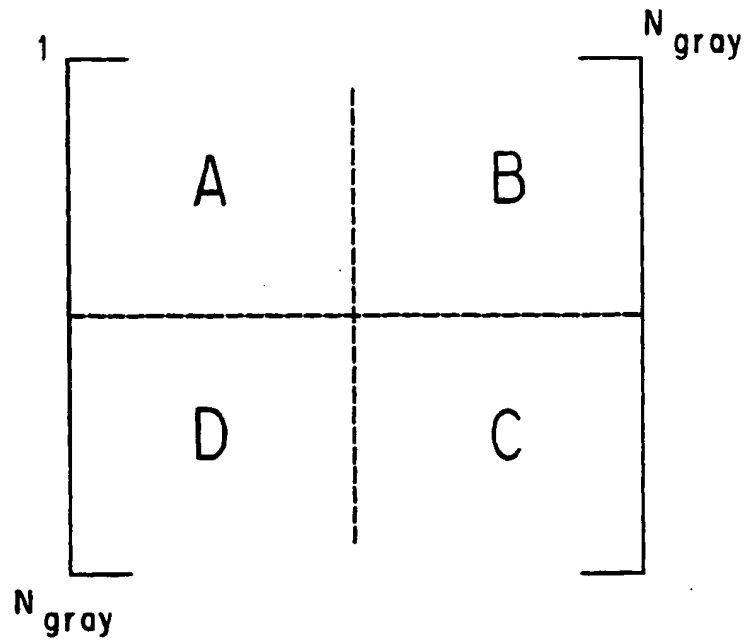


FIG.31

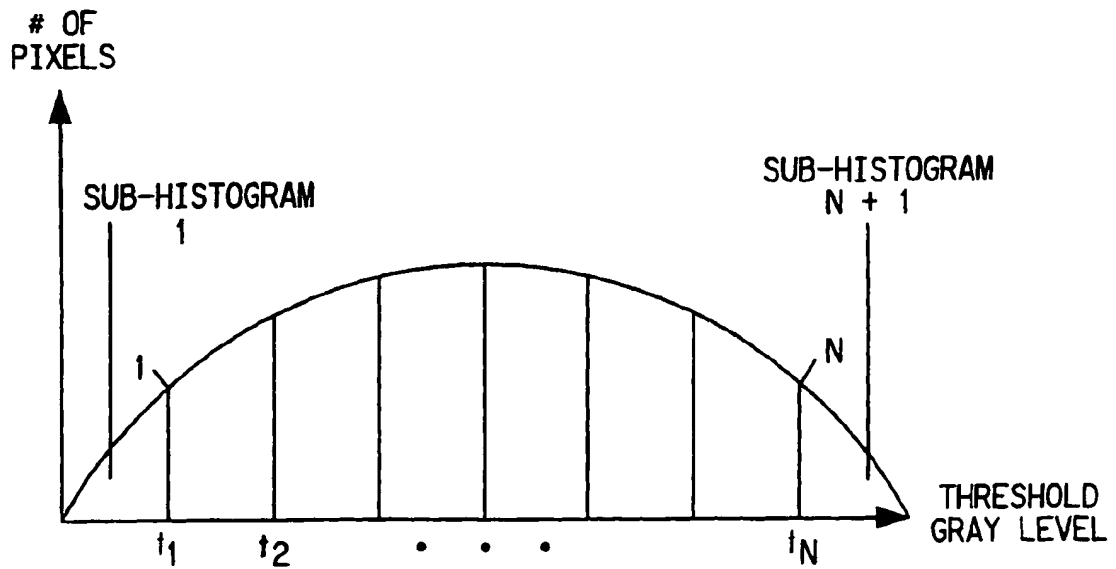


FIG. 29

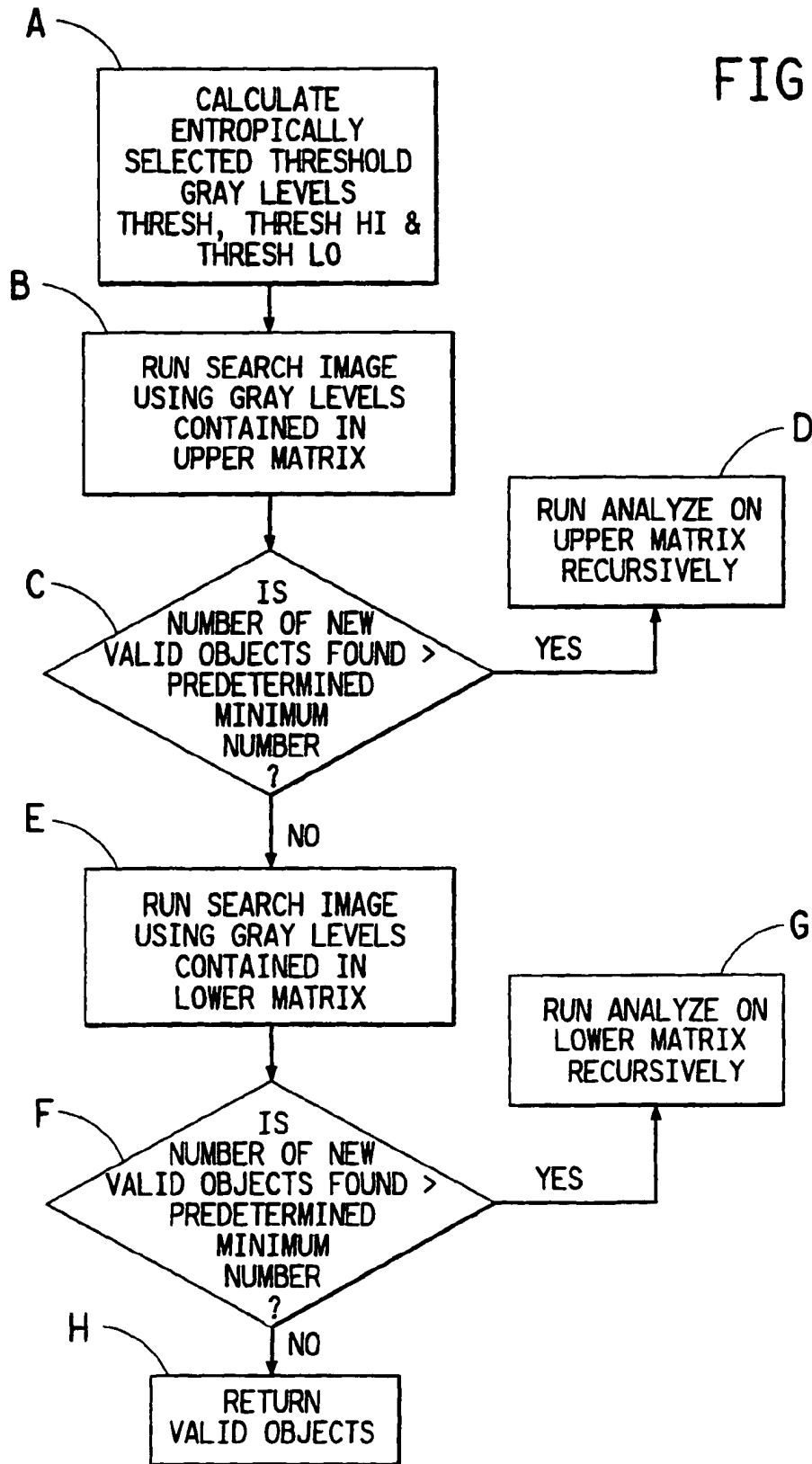


FIG. 30

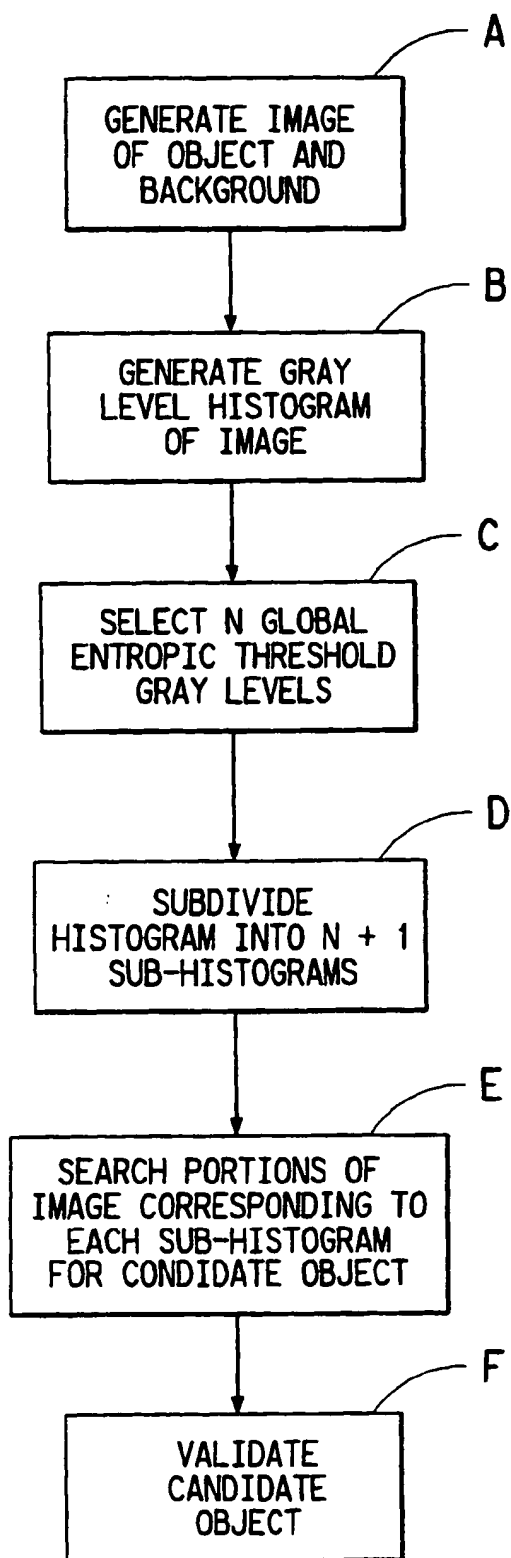


FIG. 32

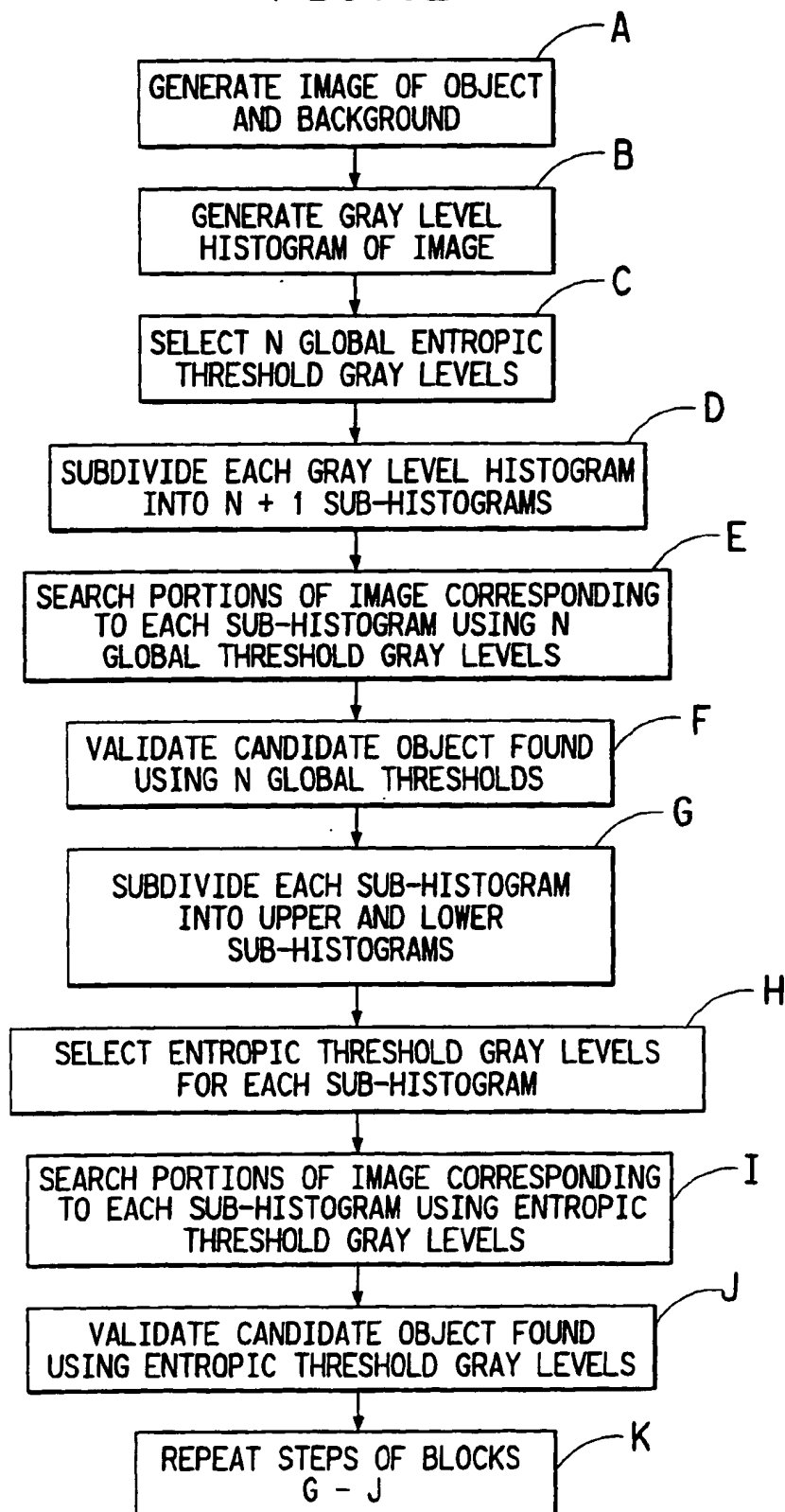


FIG. 33

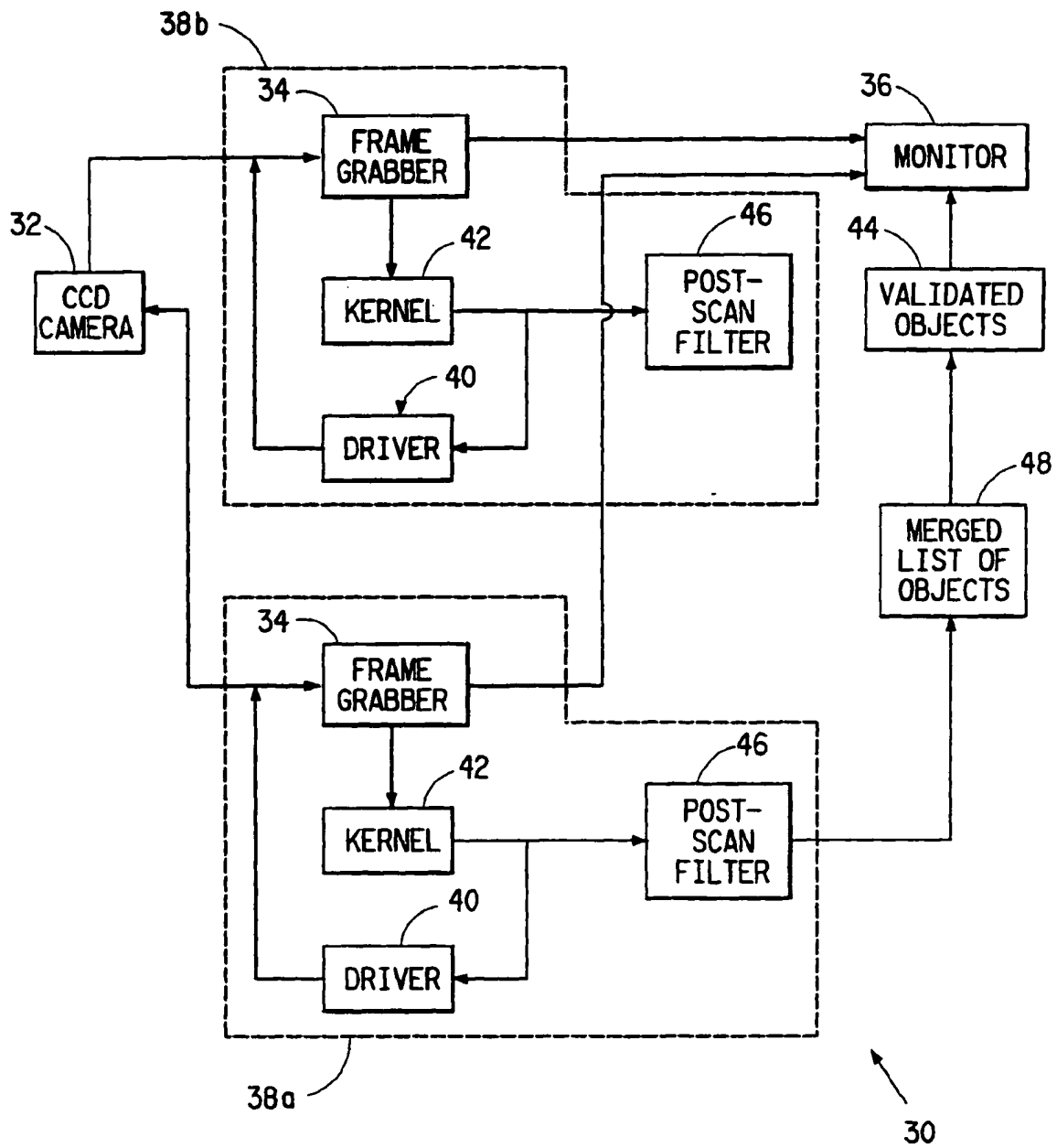


FIG. 34

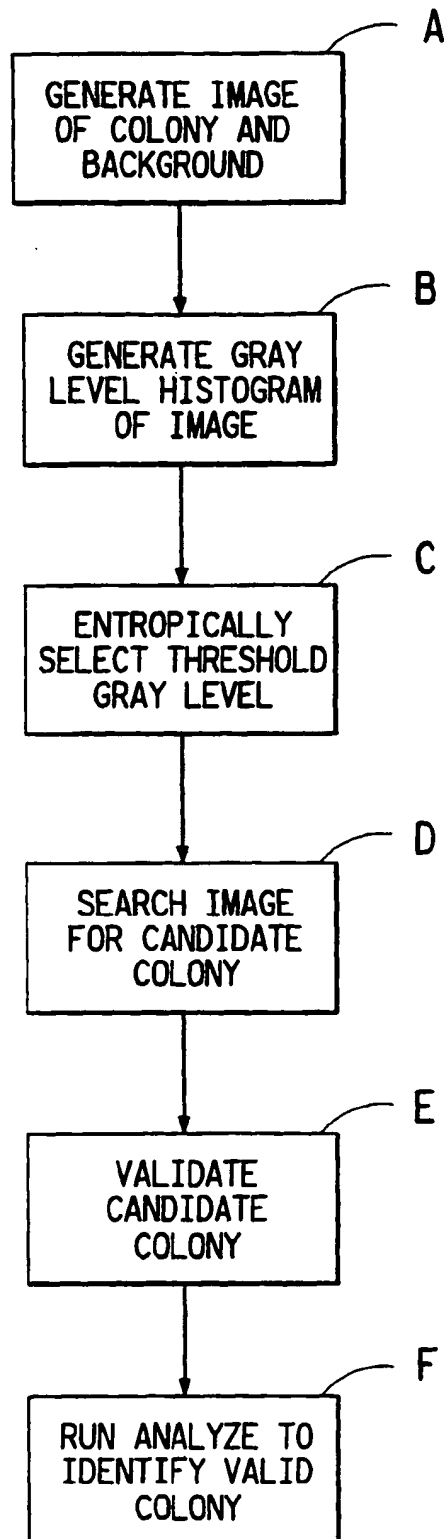


FIG. 35

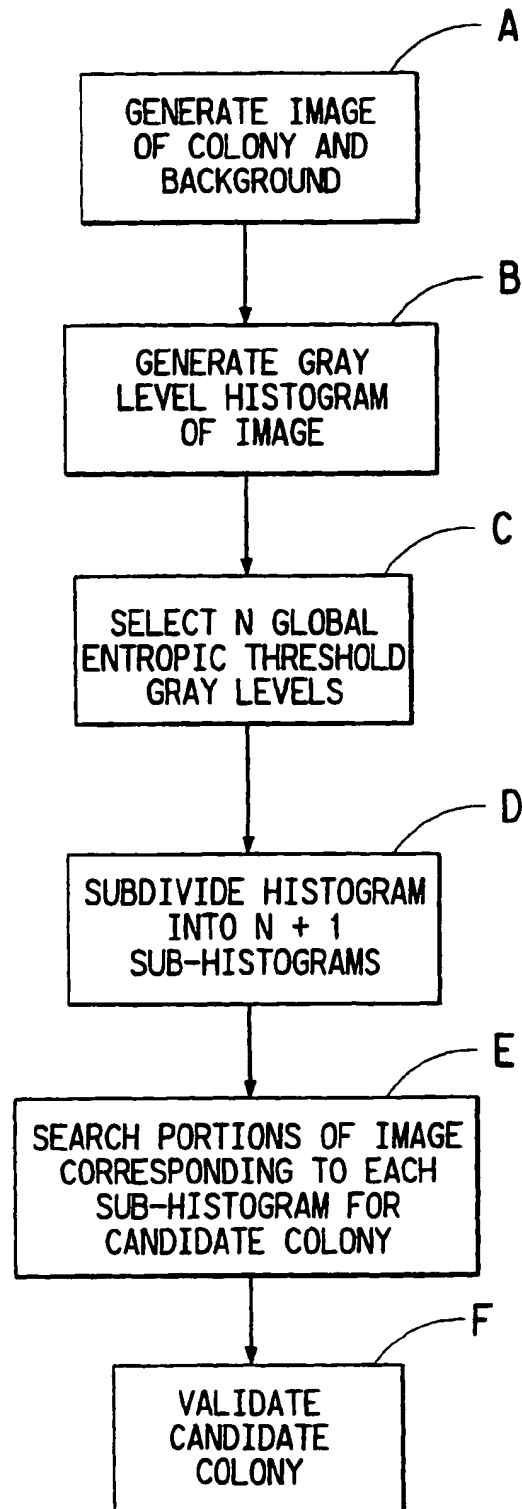


FIG. 36

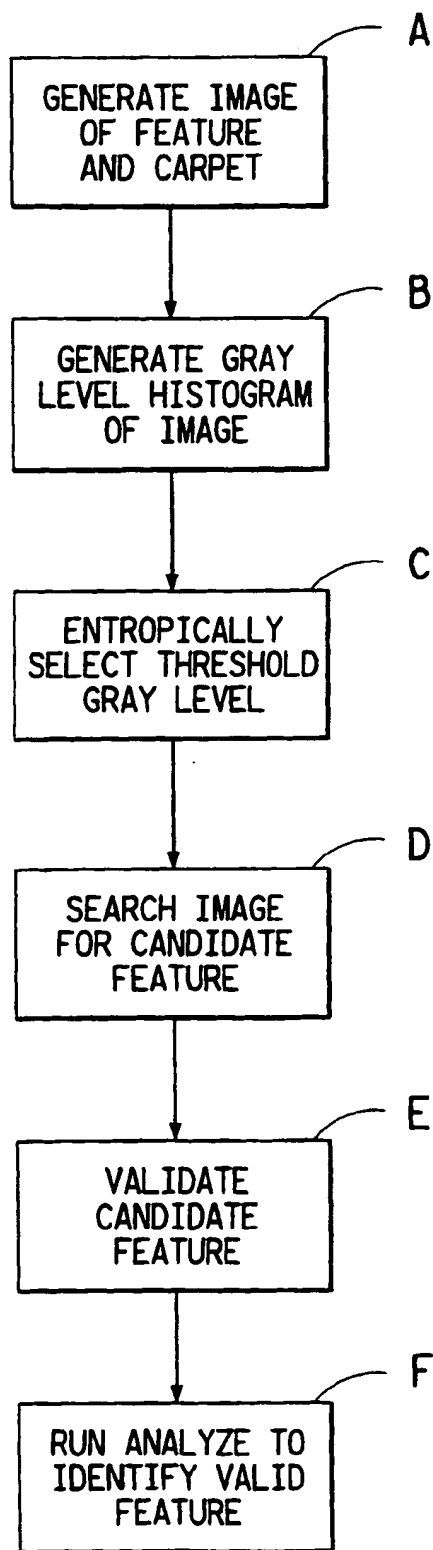


FIG. 37

